IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Mallikarjun Chadalapaka	§	Art Unit:	2152
		§		
Serial No.:	10/666,174	§		
		§	Examiner:	Thomas J. Dailey
Filed:	September 18, 2003	§		
		§		
For:	Method and Apparatus for Acknowledging a Request for Data Transfer	§	Atty. Dkt. No.:	200312982-1 (HPC.0563US)
		§		
		§		

DECLARATION UNDER 37 C.F.R. § 1.132

I, Mallikarjun Chadalapaka, state as follows:

1. I am the inventor of the subject matter of the present application (referenced above).

2. At the time of the invention of the subject matter of the present application, and at the time the present application was filed with the United States Patent and Trademark Office, I was employed by the Hewlett-Packard Co. (hereinafter "HP").

3. My current title at HP is Product Manager.

4. I conceived of the invention claimed in the present application prior to July 21, 2003. In fact, an Internal HP document dated June 11, 2003 (attached as Exhibit A) describes aspects of the invention claimed in the present application. For example, on page 3 of Exhibit A, the following statement is provided:

> Hence the proposal is to use an RDMA Read Request to "read" zero bytes out of the initiator's memory, whenever the A-bit is set on a Data-in PDU coming down from the iSCSI layer on the target. The RDMA Write/Read ordering rules of the RDMA Protocol ensure that the RDMA Read Request will not pass the RDMA Write Request and so the RDMA Read Request essentially acts to "flush" the connection of all the preceding RDMA Writes carrying the SCSI Data-in. The iSER layer on the target, when it receives the RDMA Read Response, can generate a notification back to the local iSCSI layer notifying it of the arrival of the data acknowledgment, essentially mimicking the SNACK-based data acknowledgment response.

5. Section 5 of Exhibit A further provides the following description:

Algorithm for the target iSER layer

If (the A-bit is set on the SCSI Data-in PDU) then

If (the operational ErrorRecoveryLevel=2 or if ErrorRecoveryLevel is unknown) then
Generate the standard RDMA Write for the SCSI Data-in PDU.
Generate a zero-length RDMA Read Request after the RDMA Write.
Wait for the RDMA Read Response arrival
else if (the operational ErrorRecoveryLevel=1) then
Generate the standard RDMA Write for the SCSI Data-in PDU.
Wait for the local RDMA Write Completion
endif
endif

Once the event being waited for – RDMA Read Response arrival or the local RDMA Write completion – occurs, the iSER layer on the target must generate a data acknowledgment notification to the iSCSI layer. This completes the iSCSI data acknowledgment expectations as far as the target iSCSI layer is concerned. Note that the initiator iSER or iSCSI layers need no special handling or logic in this proposed model.

6. The above cited passages of Exhibit A establish that I had possession of the claimed invention prior to July 21, 2003.

7. In at least the 2002-2003 timeframe, HP was a member of the RDMA Consortium. I (along with several other employees of HP) was a representative of HP for purposes of discussions and other activities related to the RDMA Consortium. During the 2002-2003 timeframe, I attended several meetings and participated in several telephone conference calls with other members of the RDMA Consortium for the purpose of discussing standards being considered for adoption by the RDMA Consortium.

8. In at least some of the discussions held with other members of the RDMA Consortium, I proposed that the technology covered by the subject matter of the present application be adopted by RDMA standards. As evidence of such proposal made by me, attached is an RDMAC Storage Subgroup Voting Record dated June 16, 2003 (Exhibit B). The RDMAC Storage Subgroup Voting Record was created by Mike Ko. *See* Exhibit B, p. 1. At least pages 28-30 of Exhibit B contain information proposed by me to the RDMA Consortium. Pages 28-30 of Exhibit B contain HP motions (motions made by me and others as representatives of HP) relating to iSCSI data ACKs on iSER targets. Pages 28-30 of Exhibit B contain various slides (Slide 3, Slide 1, Slide 6, and Slide 7) pertaining to data ACKs involving the iSCSI and iSER layers. The information on pages 28-30 relate to the technology covered by the present application.

9. Portions of the Technical Overview of iSCSI Extensions for RDMA (iSER) & Datamover Architecture for iSCSI (DA), by Mike Ko (Exhibit C) describe the technology that I had proposed to the RDMA Consortium. In particular, at least the following portions of Exhibit C are examples of subject matter contributed by me to the RDMA Consortium:

a. Page 8: Data_Completion_Notify notifies the iSCSI layer of the completion of inbound/outbound data transfer that was requested by the iSCSI layer when the request was qualified with Notify_Enable set (target only).

b. Page 18: If requested by the iSCSI layer during the invocation of Get_Data, the iSER layer will notify the iSCSI layer at the target using the Data_Completion_Notify Primitive upon completion of the RDMA operation.

c. Page 19: message flow diagram showing example of SCSI Write Data Transfer with acknowledge.

d. Page 21: If requested by the iSCSI layer during the invocation of Put_Data, the iSER layer will notify the iSCSI layer at the target using the Data_Completion_Notify Primitive upon completion of the RDMA Write Operation.... If the A-bit is set in the SCSI Data-in PDU, the iSER layer at the target will notify the iSCSI layer when the data transfer is complete at the initiator.... If ErrorRecoveryLevel is 2 or unknown, the iSER layer at the target will issue a zero-length RDMA Read operation and notifies the local iSCSI layer upon the completion of the RDMA Read Operation.

e. Page 24: message flow diagram showing example of SCSI Read Data Transfer with Acknowledge (A bit = 1).

10. Although several exemplary portions of Exhibit C were identified above as illustrating subject matter contributed by me to the RDMA Consortium prior to July 21, 2003, it is noted that there may be other sections of Exhibit C that were contributed by me. In general, any part of Exhibit C that relates to the following subject matter was contributed by me to the RDMA Consortium: the iSER layer receiving a request for data transfer from the iSCSI layer, determining whether the request for the data transfer contains a request for acknowledgment of completion of the data transfer; and if the request for data transfer does contain a request for acknowledgment of the completion of data transfer, waiting for an event corresponding to the completion of the request for data transfer and sending an acknowledgment to the iSCSI layer upon the occurrence of the event.

11. Thus, Mike Ko, the author of the Technical Overview in Exhibit C, derived his knowledge of the subject matter relating to the technology of the present invention and described in Exhibit C from the inventor (Mallikarjun Chadalapaka) of the present application.

12. Also, portions of the Technical Overview in Exhibit C discussed above in paragraphs 9 and 10 of this Declaration describe the work originated by the inventor (Mallikarjun Chadalapaka) of the present application.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

1/26/09
DATE

MALLIKARJUN CHADALAPAKA

Efficient realization of iSCSI Data Acknowledgment on RDMA fabrics

Mallikarjun Chadalapaka
Phone: +1 916-785-5621
Email: cbm@rose.hp.com

Networked Storage Architecture
Network Storage Solutions
Hewlett-Packard Company
Roseville CA USA

Abstract:

This paper discusses a new technique for efficient realization of iSCSI data acknowledgement requests (A-bit on Data-in PDUs) when iSCSI is operating on top of a Datamover protocol such as iSER. The traditional iSCSI data acknowledgment model involves the iSCSI layer on the target seeking a data acknowledgment from the initiator via the aforementioned A-bit, and the iSCSI layer on the initiator responding to the same via an iSCSI SNACK PDU. This traditional model, if adopted without changes to the RDMA Data movers, will lead to an extremely inefficient realization of the data acknowledgment feature in iSCSI/RDMA-Datamover implementations such as iSCSI/iSER. This paper describes a different technique that takes advantage of RDMA ordering rules and iSCSI error recovery techniques in realizing the same iSCSI data acknowledgment model in a highly optimized way.

Technical Overview of iSCSI Extensions for RDMA (iSER) & Datamover Architecture for iSCSI (DA)

RDMA Consortium

Mike Ko

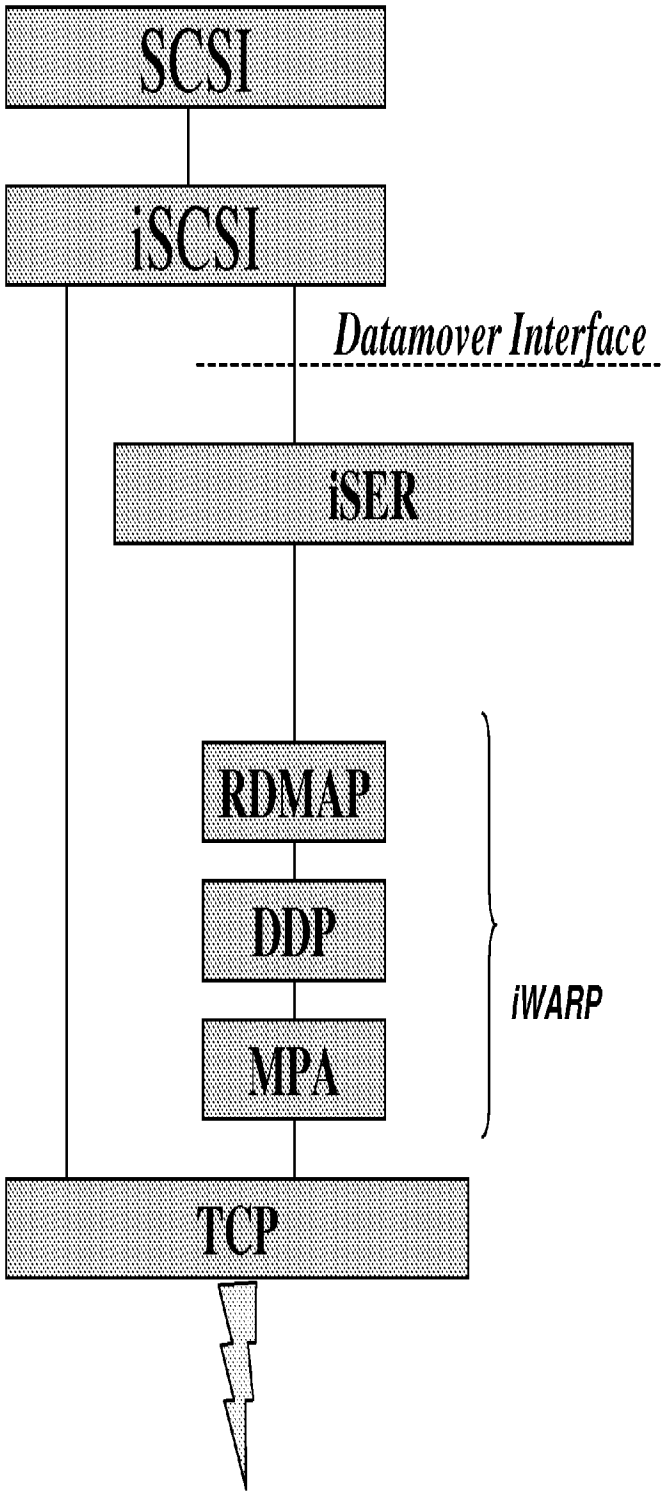
July 21, 2003

Outline

- Introduction
- iSCSI Control-type vs. Data-type PDUs
- Operational Primitives
- iSER Header
- SCSI Write Operation
- SCSI Read Operation
- Flow Control
- Connection Setup for iSER-assisted Mode
- Connection Termination for iSER-assisted Mode
- Error Recovery
- Summary

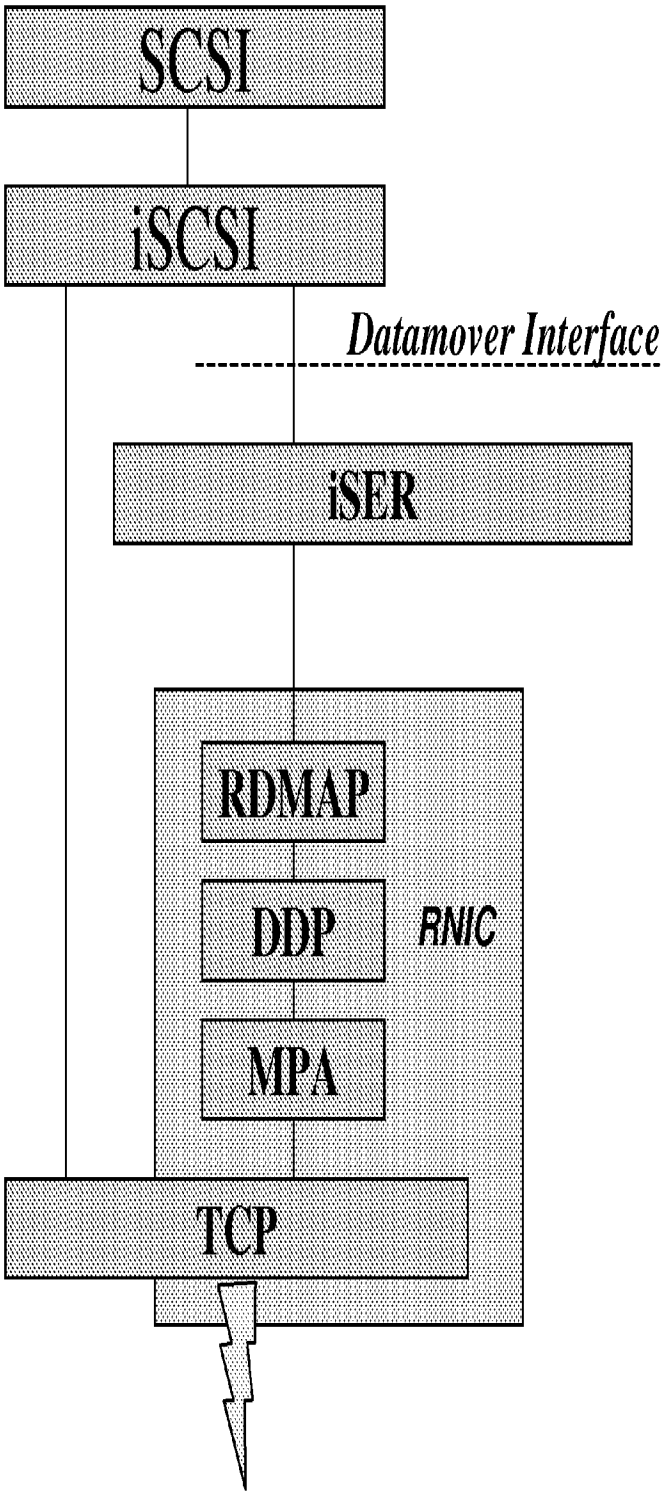
What are DA and iSER?

- The Datamover Architecture defines an abstract model in which the movement of data between iSCSI end nodes is logically separated from the rest of the iSCSI protocol
  - Allows a datamover protocol layer to offload the tasks of data movement and placement from the iSCSI layer
- The iSCSI Extensions for RDMA (iSER) protocol is one such datamover protocol
  - Applies the Datamover Architecture in extending the data transfer capabilities of iSCSI to include RDMA (Remote Direct Memory Access) as defined in the iWARP protocol suite
    - The iWARP protocol suite, submitted by the RDMA Consortium to the IETF for standardization consideration in October 2002, provides the RDMAP and DDP (Direct Data Placement) functionality to the IP fabric

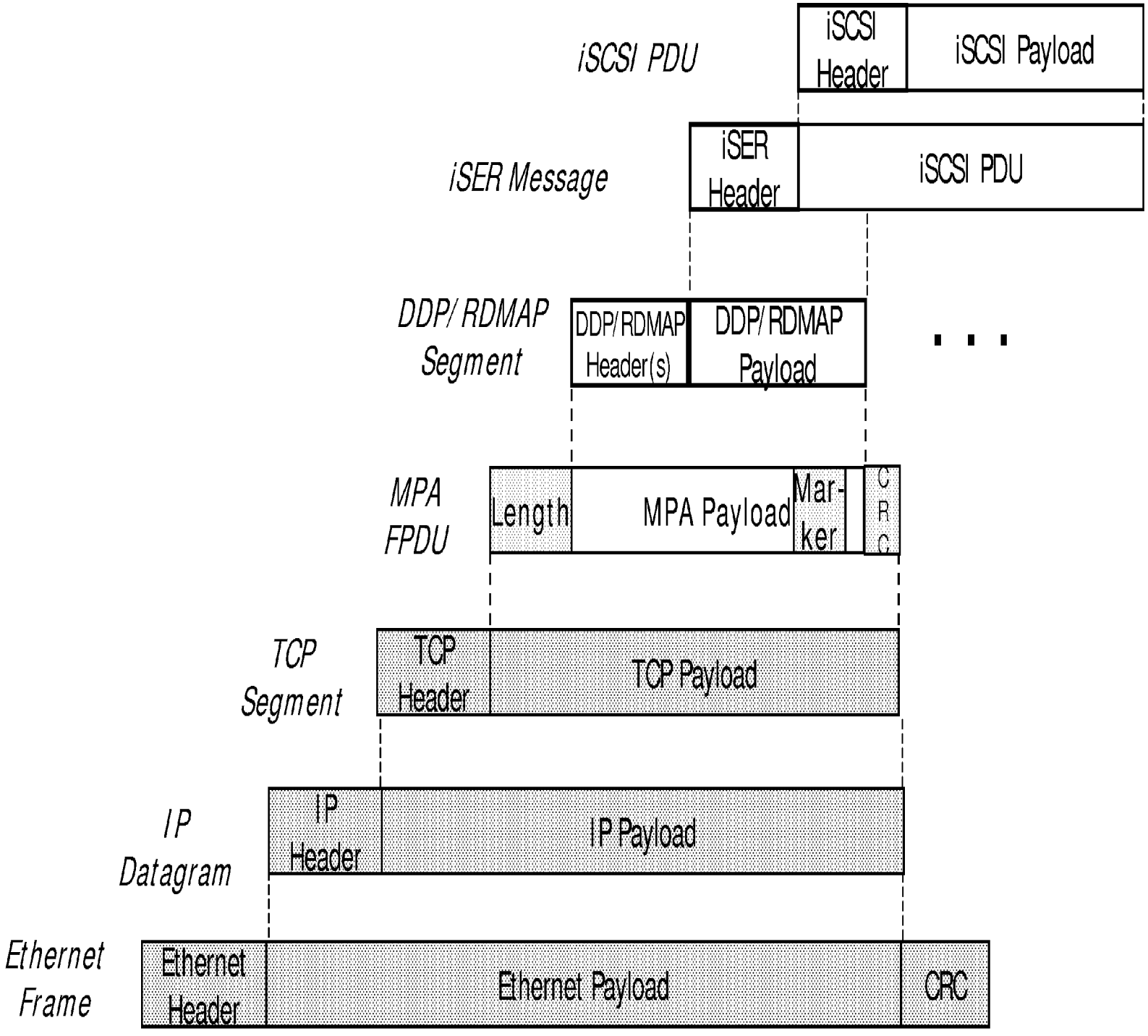


iSER & RNICs

- The iSCSI Extensions for RDMA (iSER) protocol
  - Takes advantage of the generic direct data placement mechanism and RDMA semantics offered by the iWARP technology instead of being iSCSI specific
  - Allows iSCSI implementations to have data transfers which achieve true zero copy behavior using generic RDMA network interface controllers (**RNIC**s)
    - True zero copy eliminates the increasing memory-to-memory copy overhead incurred in network protocol processing, particularly in the receive path, as speeds grow to 10 Gb/s and beyond



Example of an Encapsulation of an iSER Payload in an iWARP Message

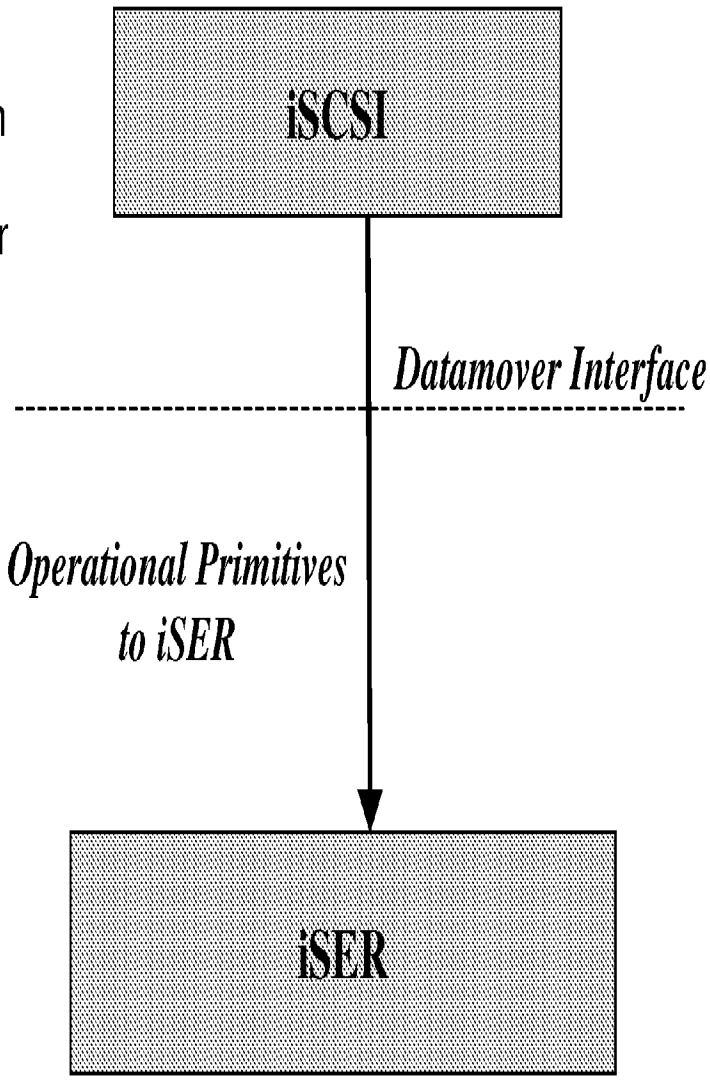


iSER Architectural Features

- iSER extends the data transfer model of the iSCSI protocol
  - Provides iWARP-based data transfer model for iSCSI that enables direct in-order or out-of-order data placement of SCSI data into pre-allocated SCSI buffers while maintaining in-order data delivery
    - Eliminates the memory-to-memory copying overhead incurred in protocol processing, particularly at the receiver, as network speeds grow to 10Gb/s and beyond
    - Requires no iSCSI or iSER specific assists in the iWARP protocol suite or RNIC
  - Simplifies certain protocol aspects of iSCSI such as data integrity management and some error recovery features
- At the same time, iSER maintains compliance with the existing iSCSI protocol
  - Requires no changes to SCSI Architecture Model (SAM/SAM-2/SAM-3) and SCSI Command set standards
  - Utilizes existing iSCSI infrastructure including but not limited to MIB, bootstrapping, negotiation, naming and discovery, and security
  - Utilizes a compatible iSCSI mechanism (login key negotiation) to determine iSER support at the initiator and the target
  - Requires a connection to continue with the semantics as defined in iSCSI if iSER is not supported by either the initiator or the target
    - Therefore, requires no full feature phase interoperability between an end node operating in iSCSI mode, and an end node operating in iSER-assisted mode

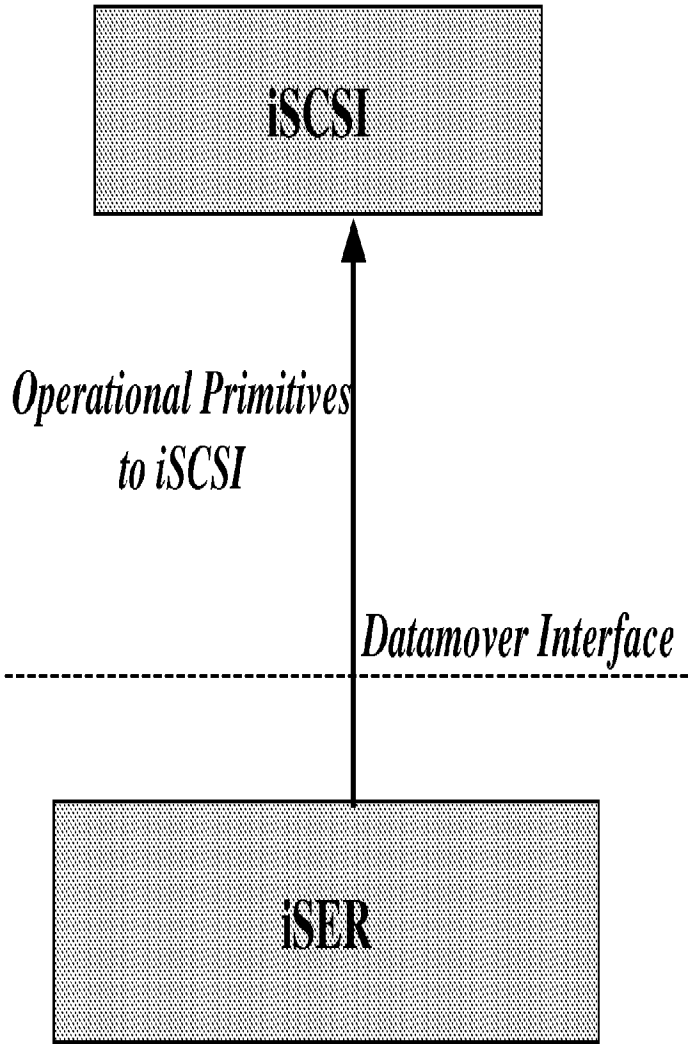
Operational Primitives Provided by the iSER Layer

- These are abstract functional interface procedures that allows the iSCSI layer to request the iSER layer to perform a specific action
- Operational Primitives that can be invoked by the iSCSI layer
  - **Send_Control** requests the outbound transfer of an iSCSI control-type PDU
  - **Put_Data** requests the outbound transfer of data for a SCSI Data-in PDU (target only)
  - **Get_Data** requests the inbound transfer of solicited data requested by an R2T PDU (target only)
  - **Allocate_Connection_Resources** requests the allocation of all iWARP-specific connection resources required for an operational iSCSI/iSER connection
  - **Deallocate_Connection_Resources** requests the deallocation of all iWARP-specific connection resources
  - **Enable_Datamover** requests that a specific iSCSI connection be transitioned to the iSER-assisted mode
  - **Connection_Terminate** requests that a specified iSCSI/iSER connection be terminated and all associated connection and task resources be freed
  - **Notice_Key_Values** requests that the specified Key-Value pairs are to be taken note of by the iSER layer
  - **Deallocate_Task_Resources** requests the deallocation of all iWARP-specific task resources



Operational Primitives Provided by the iSCSI Layer

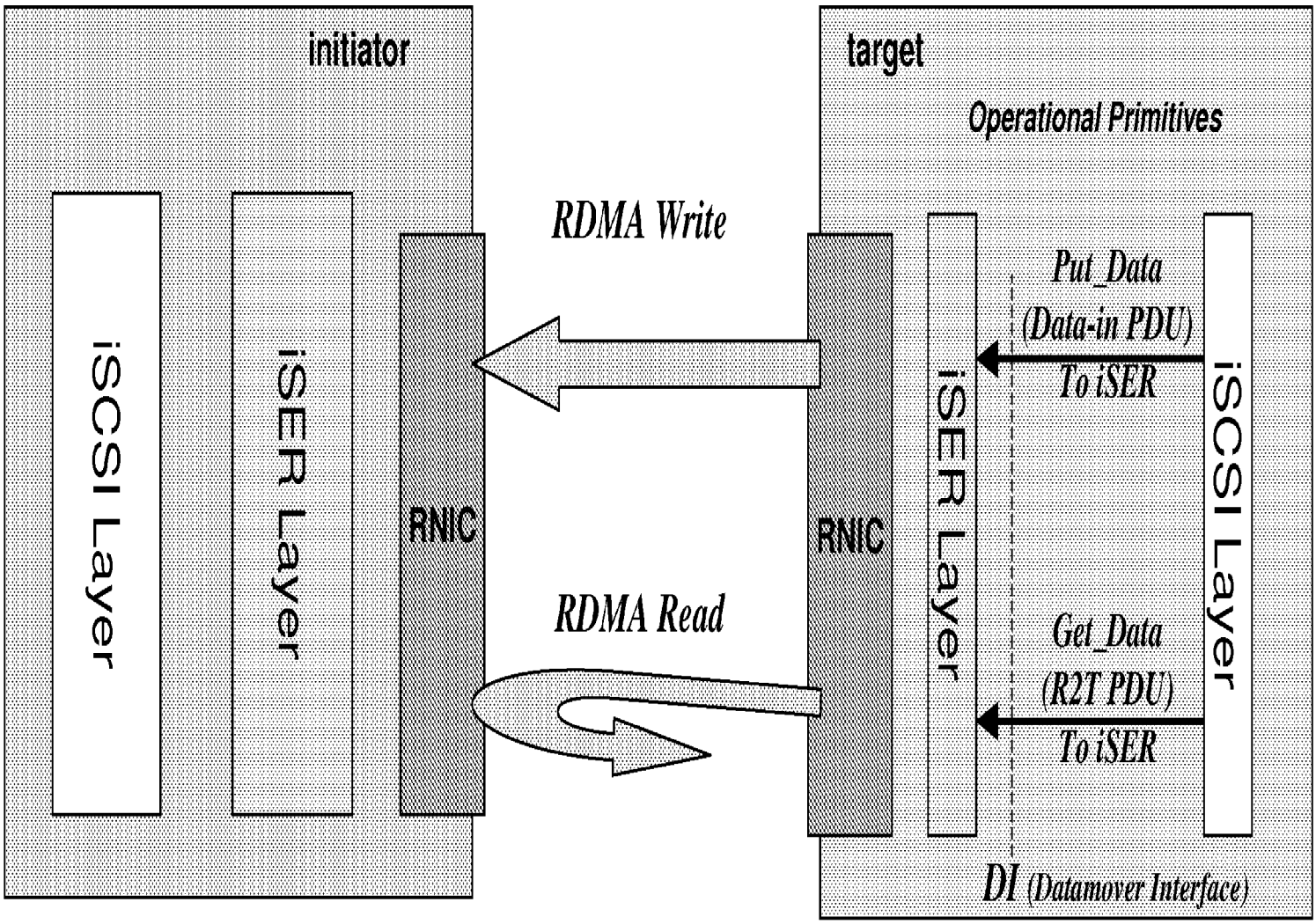
- These are abstract functional interface procedures that allows the iSER layer to notify the iSCSI layer of some event
- Operational Primitives that can be invoked by the iSER layer
  - **Control_Notify** notifies the iSCSI layer of the availability of an inbound iSCSI control-type PDU (see slide 10)
  - **Data_Completion_Notify** notifies the iSCSI layer of the completion of inbound/outbound data transfer that was requested by the iSCSI layer when the request was qualified with Notify_Enable set (target only)
  - **Data_ACK_Notify** notifies the iSCSI layer of the arrival of the data acknowledgement (target only)
  - **Connection_Terminate_Notify** notifies the iSCSI layer of the termination of an iSCSI/iSER connection



iSCSI Data -Type PDUs

- **iSCSI PDUs initiating data transfer into named buffers in the full feature phase are transformed into RDMA Read/Write Messages**
  - Defined as iSCSI data-type PDUs
  - Data transfer is managed by the RNIC hardware/firmware with no involvement from the iSCSI/iSER layers at the initiator or the target during the actual transfer
  - Include the following iSCSI PDUs only
    - **R2T** is transformed into RDMA Read operation by the target
    - **SCSI Data-in** is transformed into RDMA Write operation by the target

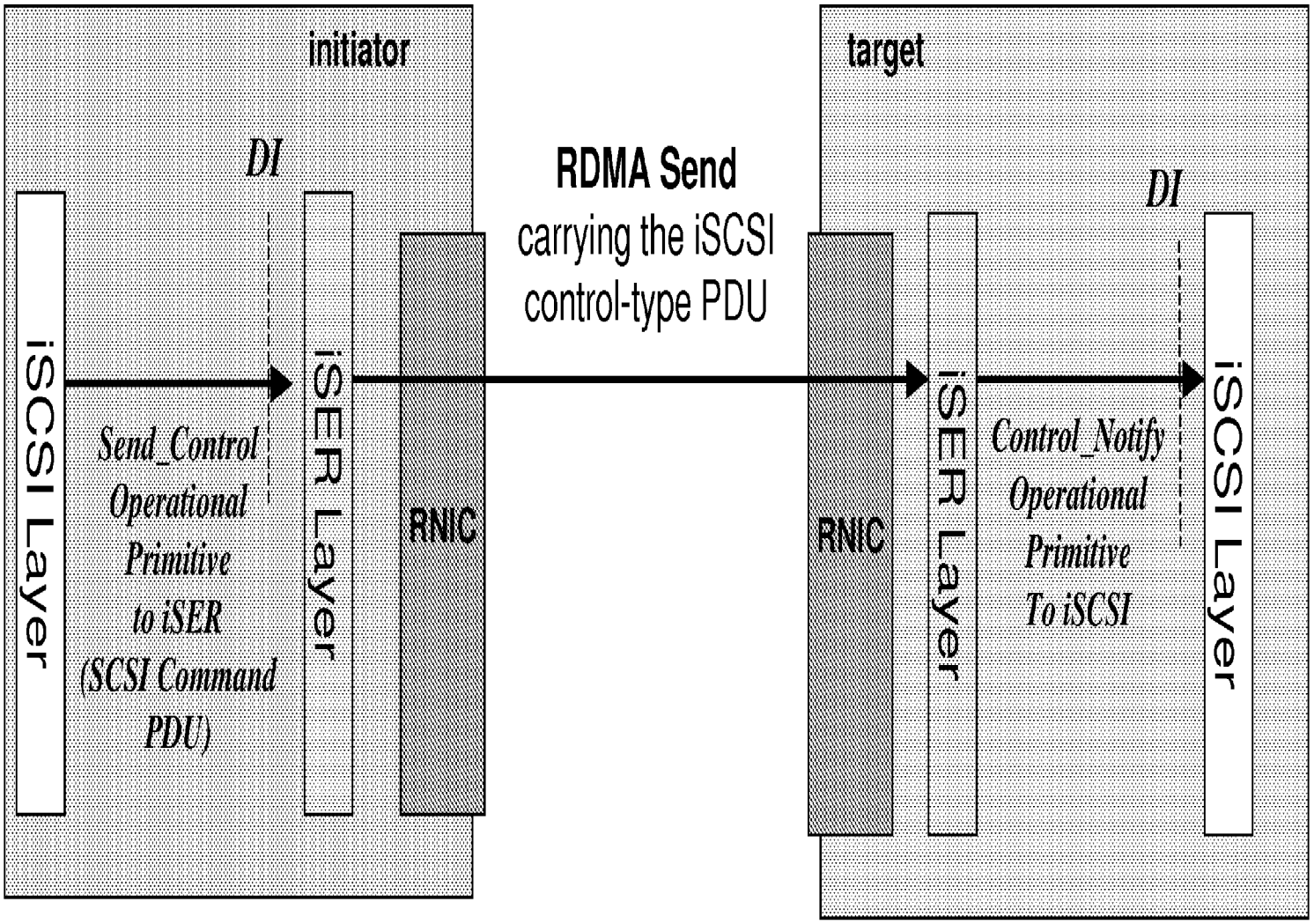
Example of iSCSI Data-Type PDUs



iSCSI Control Type PDUs

- All other iSCSI PDUs in the full feature phase are encapsulated in RDMA Send Type Messages
  - Defined as iSCSI control-type PDUs
  - iSCSI layers at the sending and the receiving nodes are involved in the PDU transfer

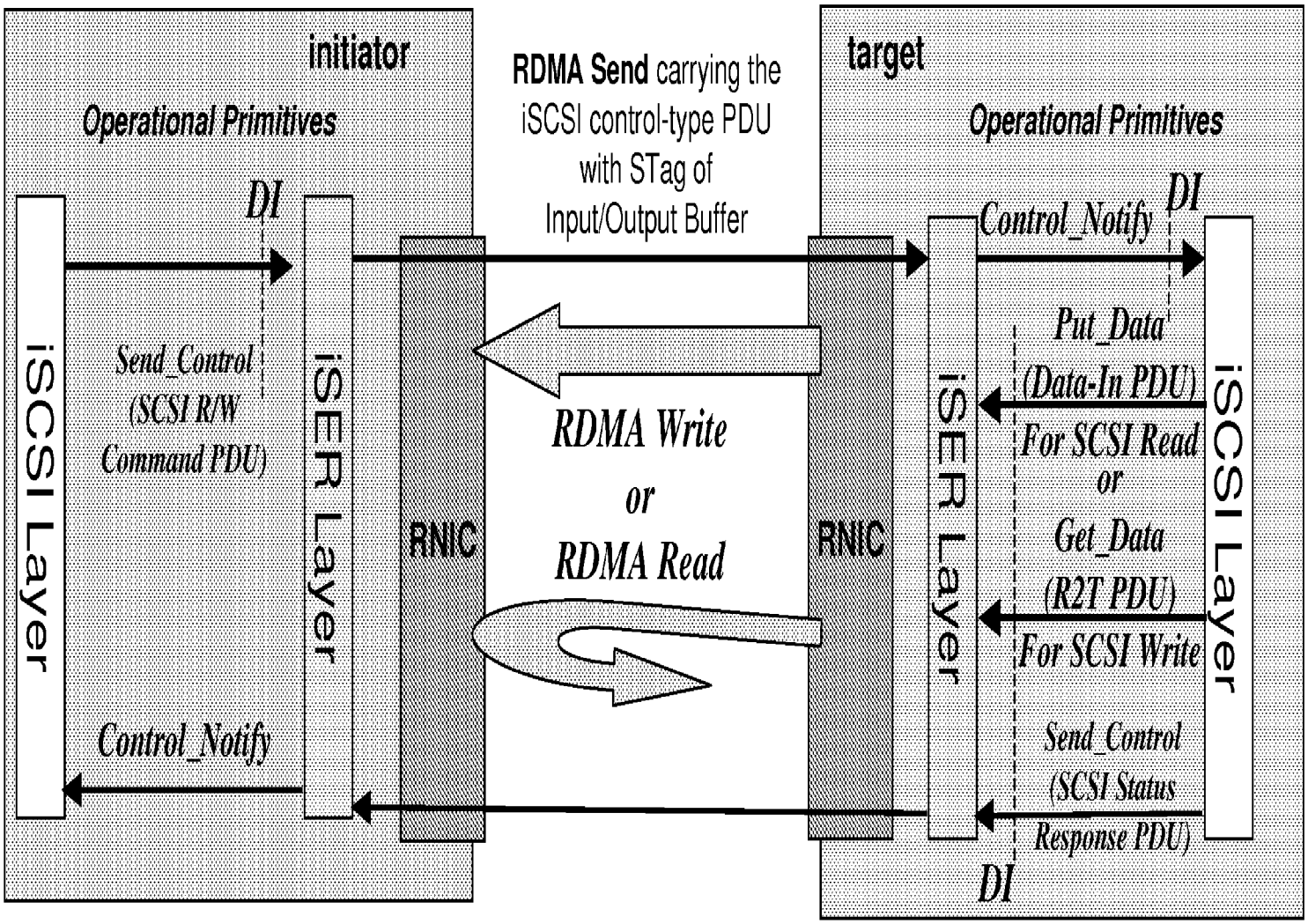
Example of sending iSCSI's SCSI Command PDU



iSER Actions for SCSI Read/Write

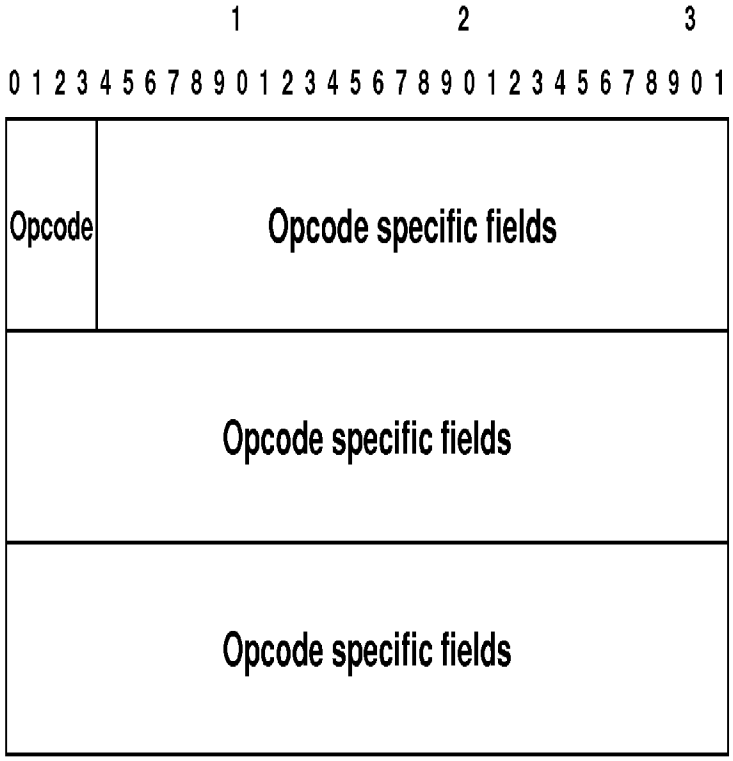
- **iSER layer at the initiator sends (advertises) the buffer identifier (STag(s)) to the target when the SCSI Command for the data-type PDU is issued by the iSCSI layer**
  - For a SCSI Read Command, the STag identifies the tagged buffer into which data from the target will be directly placed by the initiator RNIC using the RDMA Write operation
  - For a SCSI Write Command, the STag identifies the tagged buffer on the initiator from which data is directly fetched by the target RNIC using the RDMA Read operation

Example of SCSI Read or SCSI Write



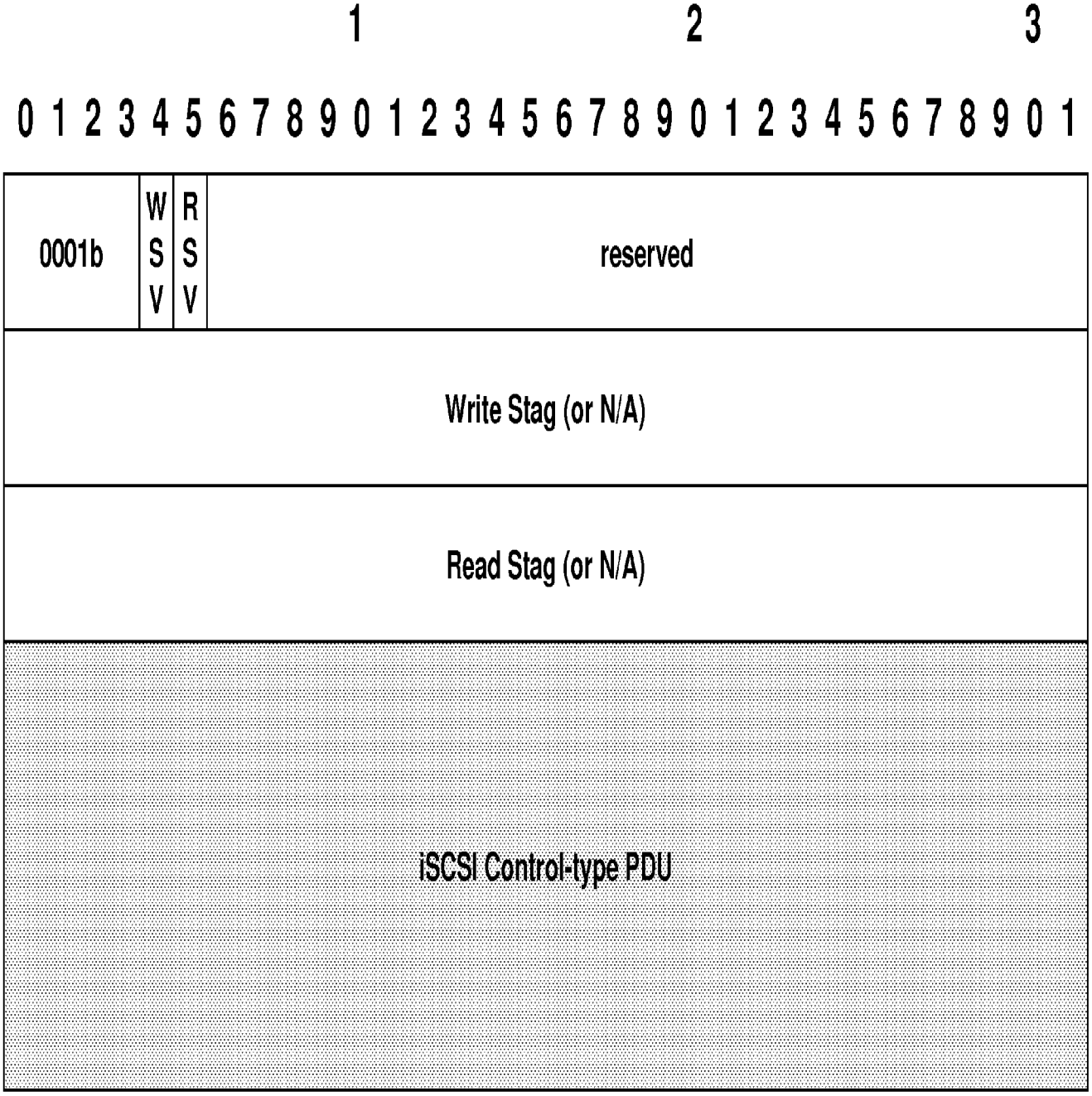
iSER Header

- The iSER header is fixed in size (12 bytes) and is present in all RDMA Send Type Messages for sending iSCSI control-type PDUs and iSER Hello/HelloReply Messages
- The iSER header serves the following purposes:
  - It provides a mechanism for the initiator to advertise the STag(s) for the tagged buffer to the target when a SCSI Command is issued by the iSCSI layer
    - The STags are used later by the target when it transforms the iSCSI data-type PDUs associated the SCSI Command into RDMA Read or RDMA Write operations
  - It allows the iSER layers at the initiator and the target to exchange operational parameters for the connection during connection setup
- The iSER header, when present, immediately follows the iWARP header (not shown)
- A 4-bit opcode field determines the layout of the iSER header



- **Opcode values:**
  - **0001b = iSCSI control-type PDU**
  - **0010b = iSER Hello Message**
  - **0011b = iSER HelloReply Message**

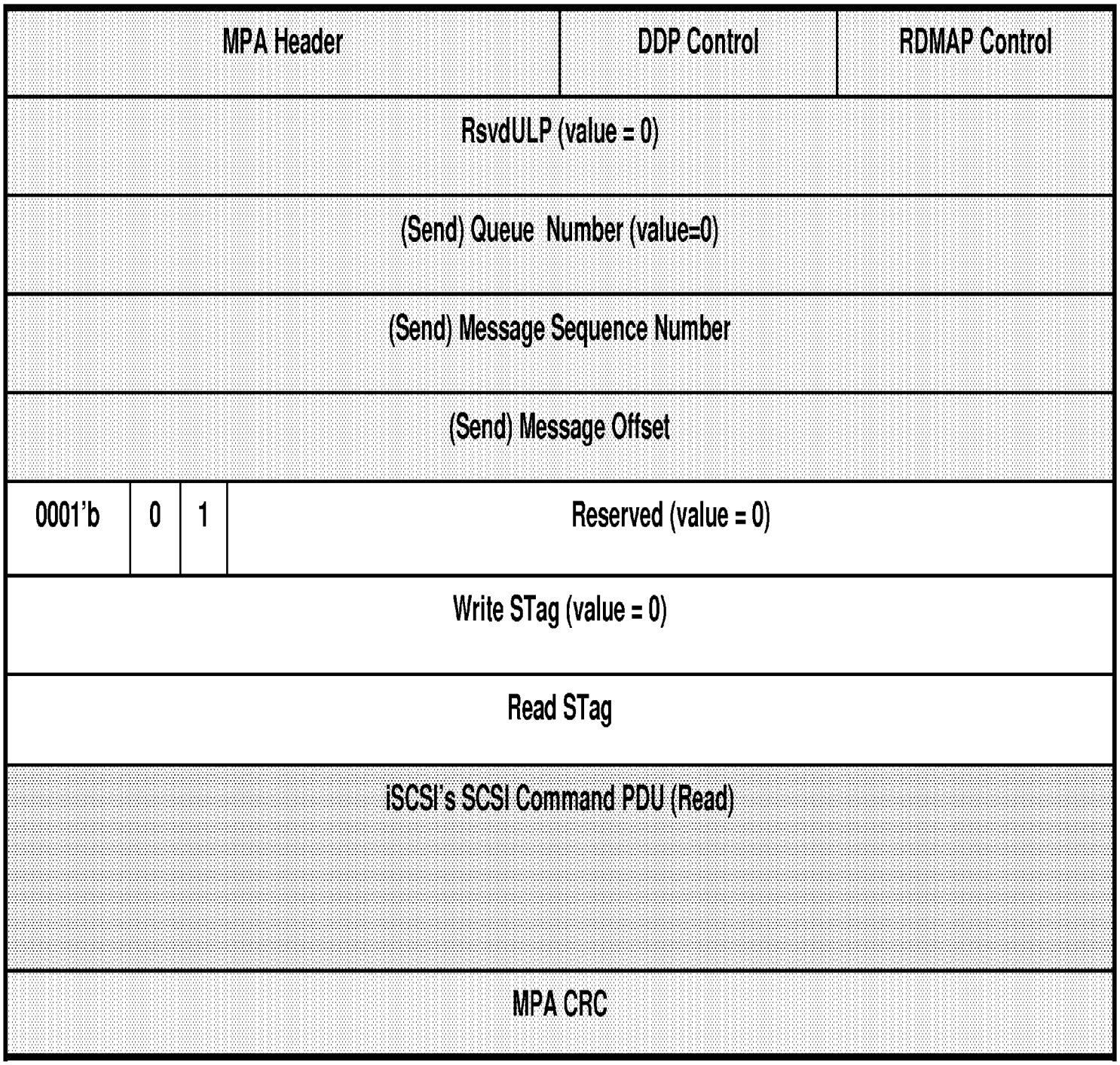
iSER Header for iSCSI control-type PDU



WSV = Write STag valid

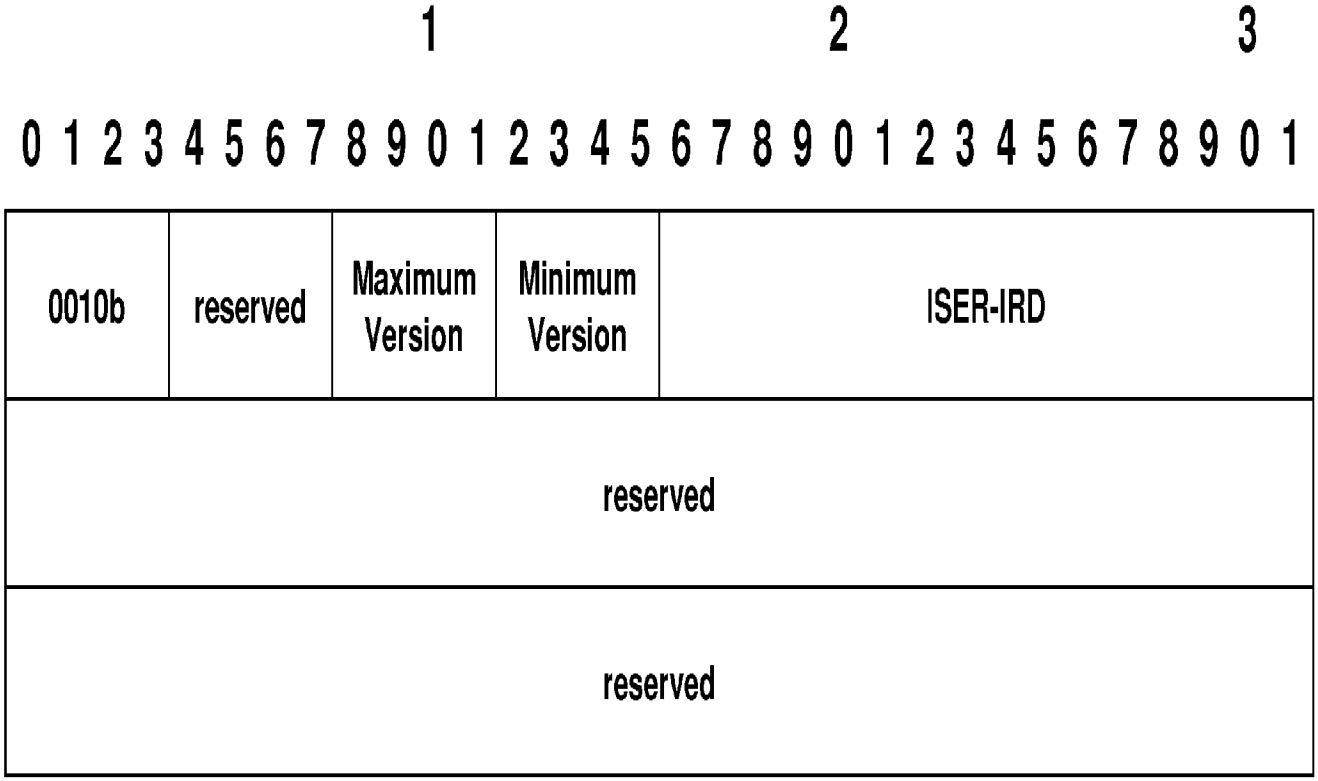
RSV = Read STag valid

Example of an iSER/iWARP Message Carrying a SCSI Command PDU of Type SCSI Read

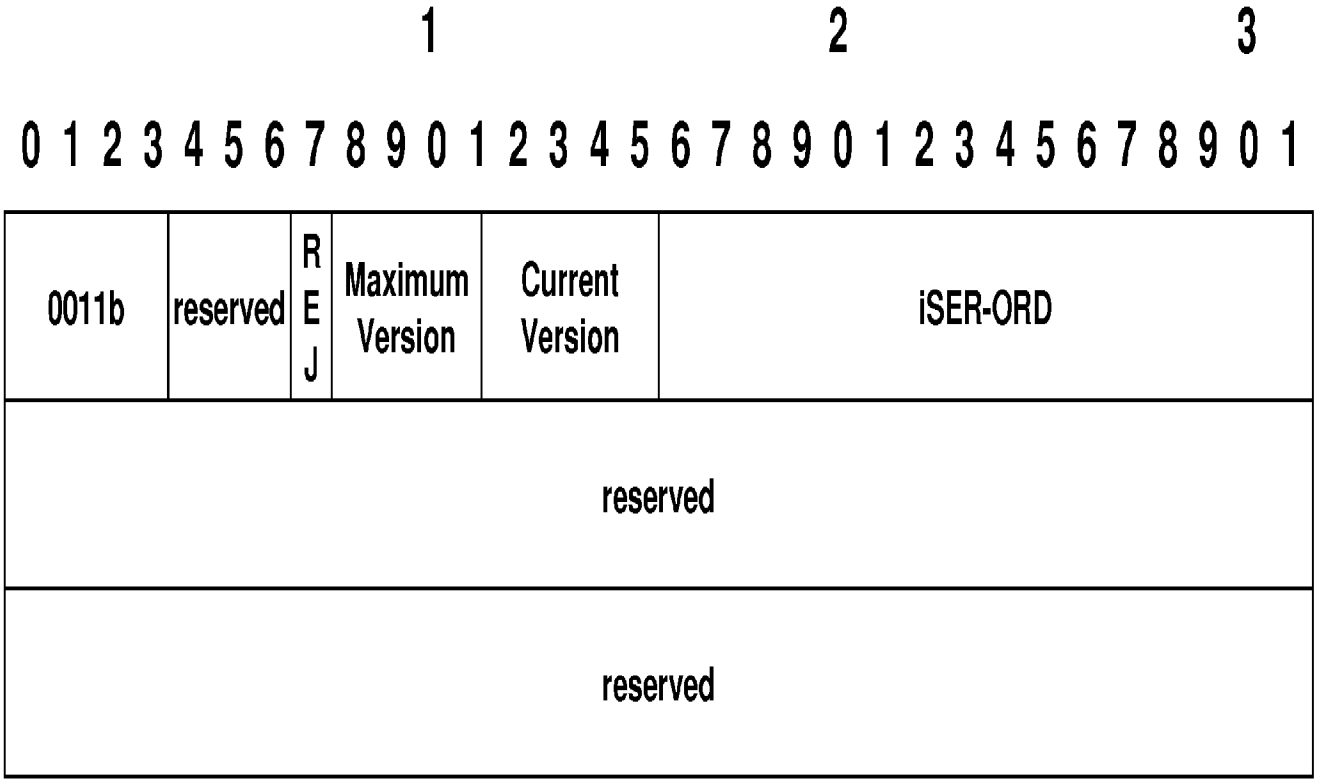


Note: Markers, if any, are not shown

iSER Header for ISER Hello Message



iSER Header for iSER HelloReply Message



- **REJ – Connection is rejected, if set to 1**

Example of an iSER Hello Message in an iWARP Send Message

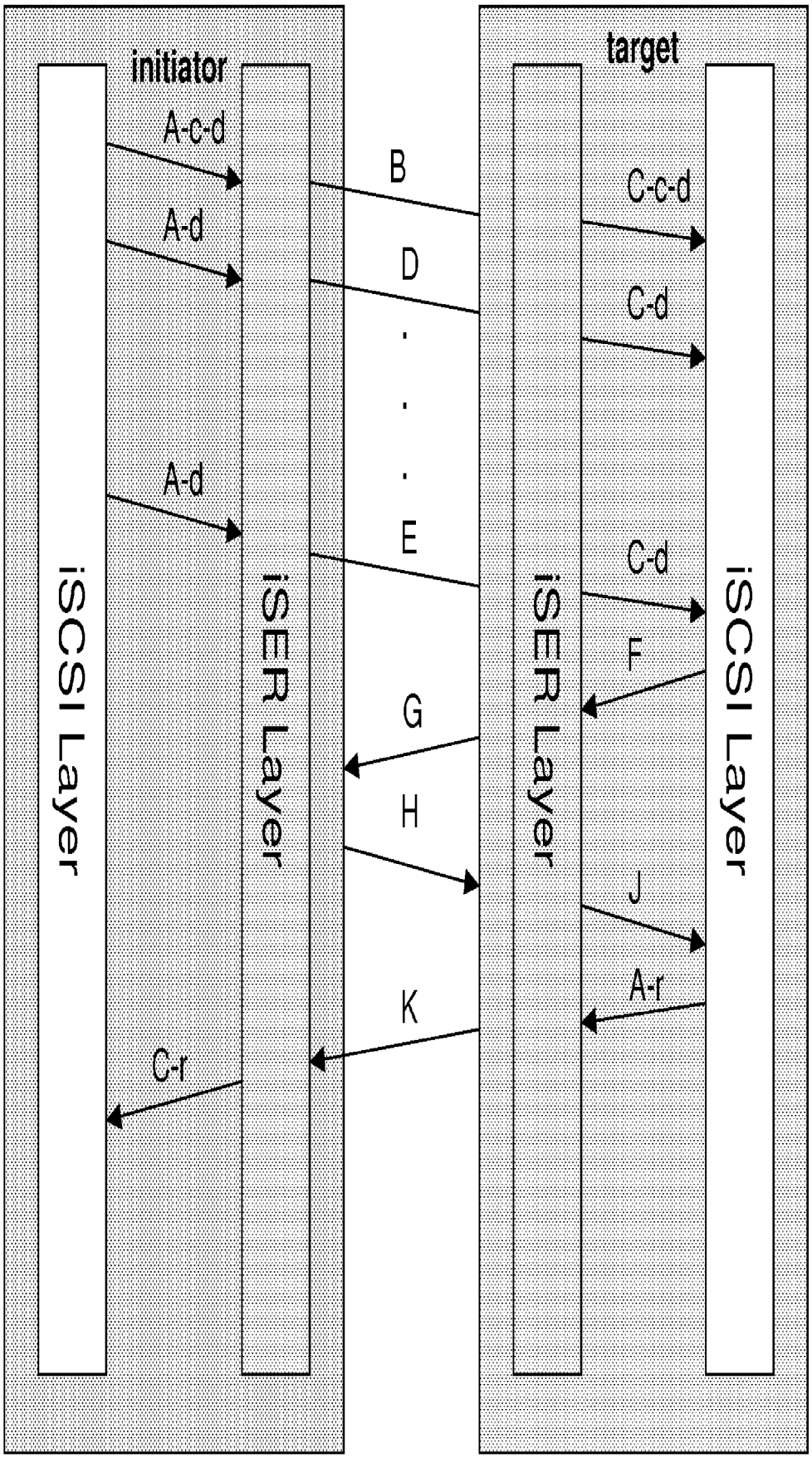
<table>
<tr><td colspan="4">MPA Header</td><td>DDP Control</td><td>RDMAP Control</td></tr>
<tr><td colspan="6">RsvdULP (value = 0)</td></tr>
<tr><td colspan="6">(Send) Queue Number (value=0)</td></tr>
<tr><td colspan="6">(Send) Message Sequence Number</td></tr>
<tr><td colspan="6">(Send) Message Offset</td></tr>
<tr><td>0010’b</td><td>Reserved
(values = 0)</td><td>0001’b</td><td>0001’b</td><td colspan="2">iSER-IRD</td></tr>
<tr><td colspan="6">Reserved (value = 0)</td></tr>
<tr><td colspan="6">Reserved (value = 0)</td></tr>
<tr><td colspan="6">MPA CRC</td></tr>
</table>

Note: Markers, if any, are not shown

SCSI Write Operation

- The iSCSI layer at the initiator invokes the Send_Control Primitive to request the iSER layer to send the SCSI Command PDU
  - The iSER layer requests the RDMAP layer to transmit a Send Message containing the SCSI Command PDU and immediate data (if any)
  - If there is solicited data, the iSER layer at the initiator advertises the Write STag in the iSER header in the Send Message
- Upon receiving the Send Message containing the SCSI Command PDU, the iSER layer at the target notifies the iSCSI layer using the Control_Notify Primitive
- If there is non-immediate unsolicited data, the iSCSI layer at the initiator invokes the Send_Control Primitive to request the iSER layer to send the SCSI Data-out PDU
- Upon receiving the Send Message containing the SCSI Data-out PDU, the iSER layer at the target will notify the iSCSI layer using the Control_Notify Primitive
- If there is solicited data, the iSCSI layer at the target invokes the Get_Data Primitive when it has an I/O buffer available to request the iSER layer to handle the R2T PDU
  - The iSER layer at the target transforms each R2T into an RDMA Read Operation
- If requested by the iSCSI layer during the invocation of Get_Data, the iSER layer will notify the iSCSI layer at the target using the Data_Completion_Notify Primitive upon the completion of the RDMA operation
- Upon completing the data transfer, the iSCSI layer at the target invokes the Send_Control Primitive to request the iSER layer to send the SCSI Response PDU. The iSER layer passes the STag in the RDMA Send with Invalidate Message.
- Upon receiving the Send with Invalidate Message containing the SCSI Response PDU, the RNIC at the initiator invalidates the STag and notifies the iSCSI layer of the iSCSI PDU using the Control_Notify Primitive

Example of SCSI Write Data Transfer



A. Send_Control to send SCSI PDU
   c - command
   d -data
   r - response

B. iWARP Send Message containing SCSI Command PDU with immediate data

C. Control_Notify to report SCSI PDU received
   c – command
   d – data
   r – response

D. iWARP Send Message containing SCSI Data-out PDU with unsolicited data

E. iWARP Send Message containing SCSI Data-out PDU with last unsolicited data segment

F. Get_Data for R2T

G. iWARP RDMA Read Request (*)

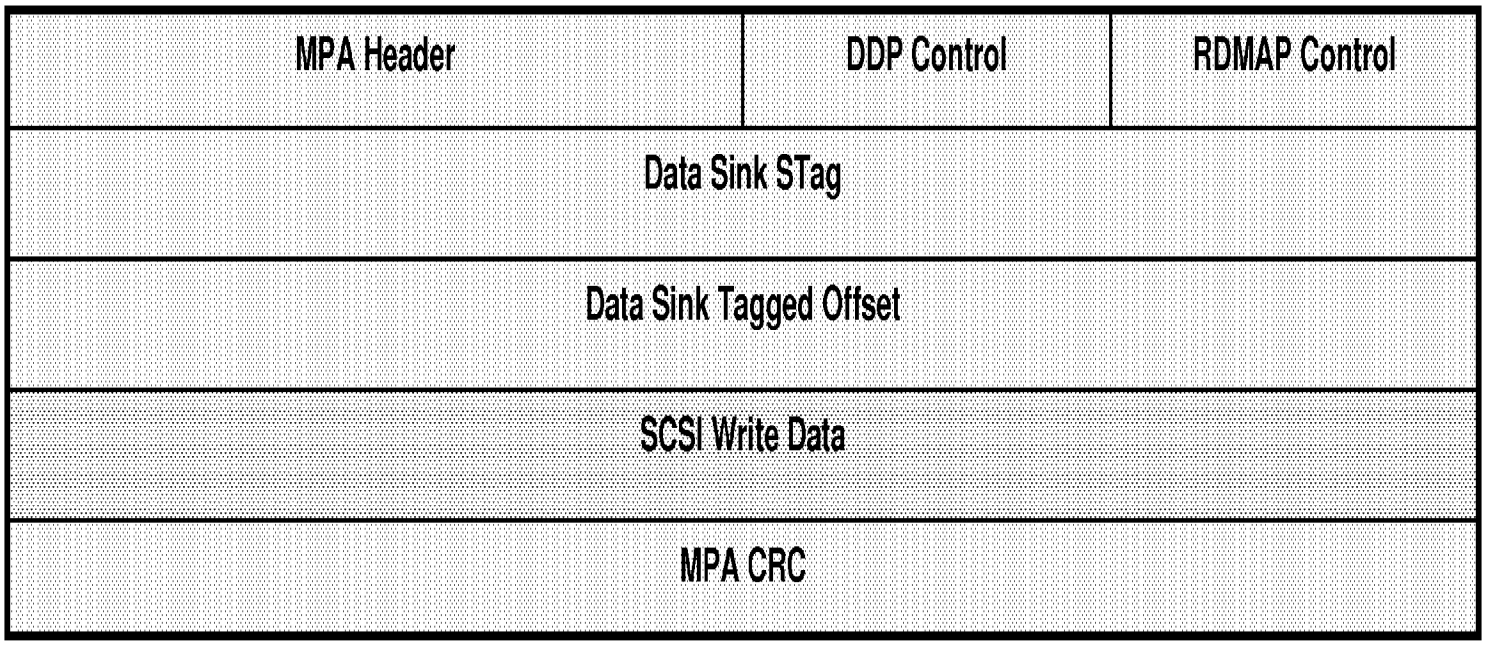
H. iWARP RDMA Read Response with solicited data (*)

J. Data_Completion_Notify

K. iWARP Send with Invalidate Message containing SCSI Response PDU

* RDMA Read data transfer is handled by the RNIC

Example of an RDMA Read Response Message Containing SCSI Write Data

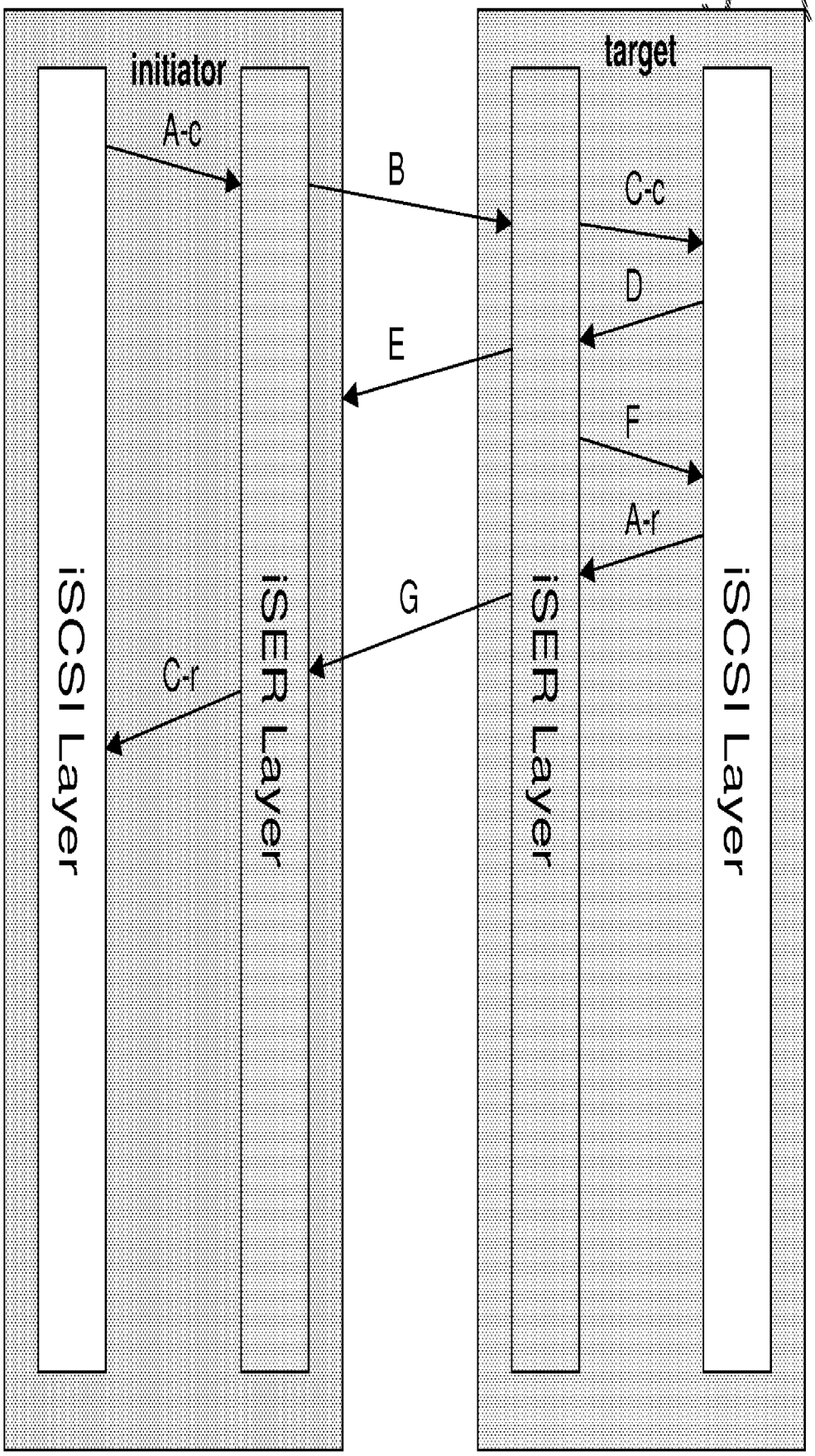


Note: Markers, if any, are not shown

SCSI Read Operation

- The iSCSI layer at the initiator invokes the Send_Control Primitive to request the iSER layer to send the SCSI Read Command PDU.
- The iSER layer requests the RDMAP layer to transmit a Send Message containing the SCSI Command PDU
  - The iSER layer advertises the Read STag in the iSER header in the Send Message
- Upon receiving the Send Message containing the SCSI Command PDU, the iSER layer at the target notifies the iSCSI layer using the Control_Notify Primitive
- When the requested data is available at the I/O buffer, the iSCSI layer at the target invokes the Put_Data Primitive to request the iSER layer to handle the SCSI Data-in PDU
  - The iSER layer at the target transforms the SCSI Data-in PDU into an RDMA Write operation
- If requested by the iSCSI layer during the invocation of Put_Data, the iSER layer will notify the iSCSI layer at the target using the Data_Completion_Notify Primitive upon the completion of the RDMA Write operation
- If the A-bit is set in the SCSI Data-in PDU, the iSER layer at the target will notify the iSCSI layer when the data transfer is complete at the initiator
  - If ErrorRecoveryLevel is 2 or unknown, the iSER layer at the target will issue a zero-length RDMA Read operation and notifies the local iSCSI layer upon the completion of the RDMA Read operation
- Upon completing the data transfer, the iSCSI layer at the target invokes the Send_Control Primitive to request the iSER layer to send the SCSI Response
  - SCSI status is always returned in a separate SCSI Response PDU ("Phase collapse" in SCSI Read Command is not allowed)
  - The iSER layer passes the STag in the RDMA Send with Invalidate Message
- Upon receiving the Send with Invalidate Message containing the SCSI Response PDU, the RNIC at the initiator invalidates the STag and the iSER layer notifies the local iSCSI layer using the Control_Notify Primitive

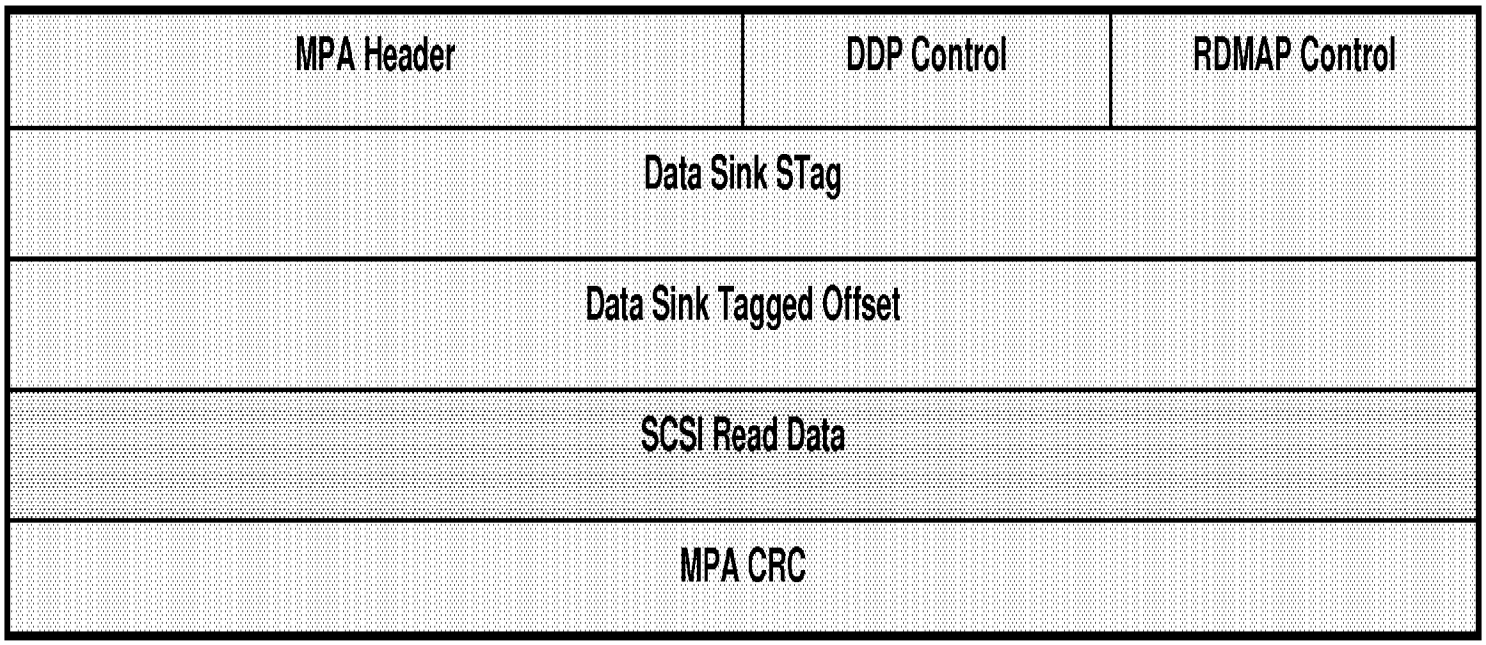
Example of SCSI Read Data Transfer with no Acknowledge (A bit = 0)



A. Send_Control to send SCSI PDU
   c – command
   r - response
B. iWARP Send Message containing SCSI Command PDU
C. Control_Notify to report SCSI PDU received
   c – command
   r - response
D. Put_Data for SCSI Data-in PDU
E. iWARP RDMA Write Message (*)
F. Data_Completion_Notify
G. iWARP Send with Invalidate Message containing SCSI Response PDU

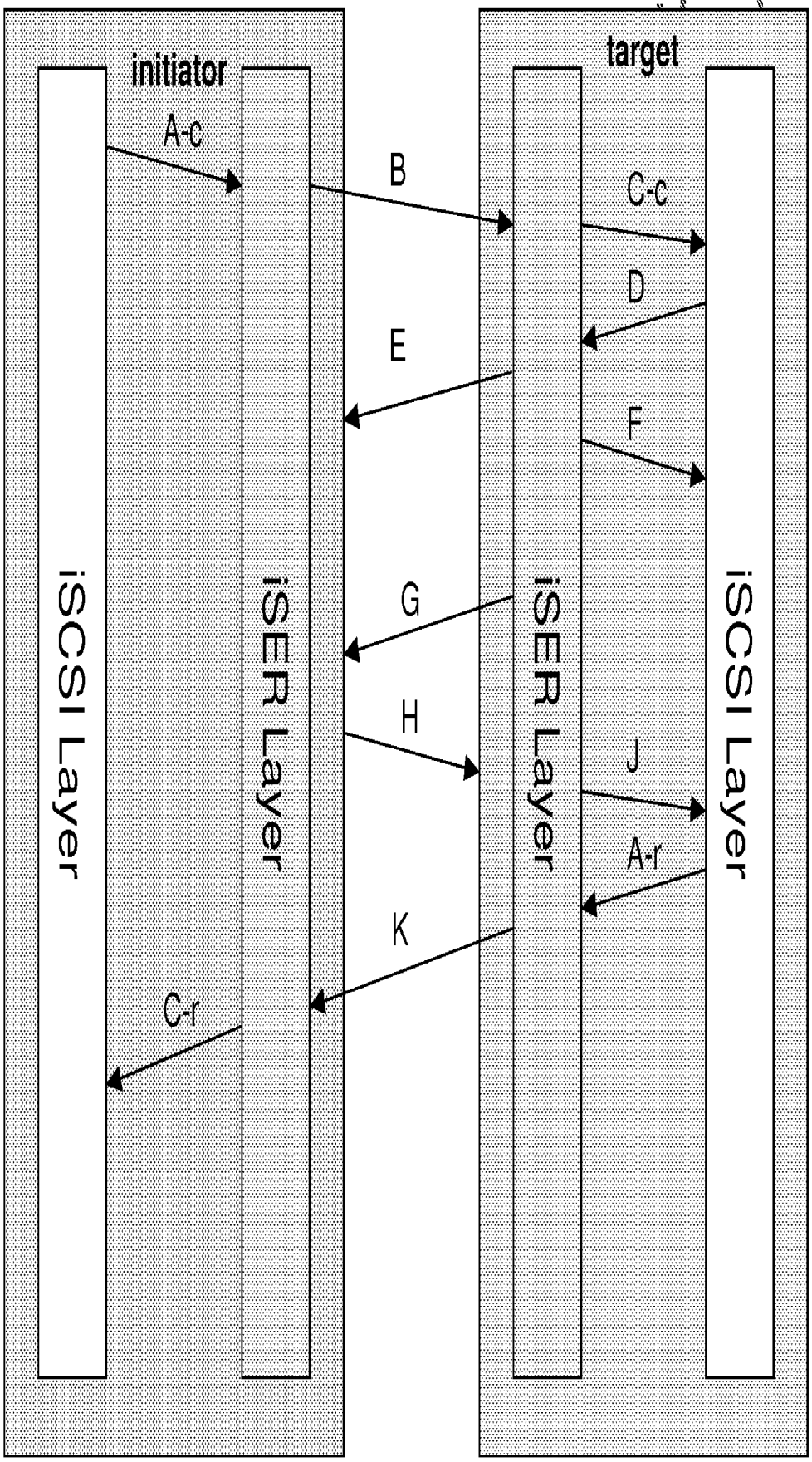
* RDMA Write data transfer is handled by the RNIC

Example of an RDMA Write Message Containing SCSI Read Data



Note: Markers, if any, are not shown

Example of SCSI Read Data Transfer with Acknowledge (A bit = 1)



A. Send_Control to send SCSI PDU
   c – command
   r - response
B. iWARP Send Message containing SCSI Command PDU
C. Control_Notify to report SCSI PDU received
   c – command
   r - response
D. Put_Data to process SCSI Data-in PDU
E. iWARP RDMA Write Message (*)
F. Optional Data_Completion_Notify
G. iWARP RDMA Read Request with zero length
H. iWARP RDMA Read Response with zero length
J. Data_ACK_Notify to report data received at initiator
K. iWARP Send with Invalidate Message containing SCSI Response PDU

* RDMA Write data transfer is handled by the RNIC

Flow Control

- For RDMA Send Type Messages
  - The iSER protocol does not provide additional flow control beyond that provided by the iSCSI layer on control-type PDUs
  - An implementation should be able to take advantage of iWARP Verbs mechanisms such as the Shared Receive Queue mechanism to effectively address the Send Message flow control question
- For RDMA Read Resources
  - In the iSER Hello Message, the iSER layer at the initiator declares the maximum number of RDMA Read Requests that the initiator can receive on the particular RDMAP Stream (iSER-IRD) to the target
    - This allows the iSER layer at the target to adjust its resources if it can issue more RDMA Read Requests than the initiator can handle
  - In the iSER HelloReply Message, the iSER layer at the target declares the maximum number of RDMA Read Requests that the target can issue on a particular RDMAP Stream (iSER-ORD) to the initiator
    - This allows the iSER layer at the initiator to adjust its resources if it can handle more RDMA Read Requests than the target can issue
  - The iSER layer at the target will flow control the RDMA Read Request Messages to not exceed iSER-ORD

General Considerations for an iSCSI/iSER Connection Setup

- During connection setup, the iSCSI layer at the initiator is responsible for establishing a TCP connection
  - Use the TCP port as discovered through the iSCSI discovery mechanisms
- iSCSI Login negotiation follows the same rules as in the iSCSI specification with the following changes:
  - The iSCSI layers at the initiator and the target negotiate the new RDMAExtensions key on the leading connection in order to enable iSER-assisted mode
  - Header Digest and Data Digests are negotiated to "None"
    - Data integrity is already provided by the MPA CRC
    - Managing the digests in the RNIC would mean that the RNIC have to be ULP-aware
  - The iSCSI layer negotiates the new TargetRecvDataSegmentLength key and the InitiatorRecvDataSegmentLength key for each connection
    - Whenever the initiator (or target) sends an iSCSI control-type PDU to the target (or initiator), non-final PDUs have a data segment size of exactly TargetRecvDataSegmentLength (or InitiatorRecvDataSegmentLength)
  - OFMarker and IFMarker are negotiated to "No"
    - Markers are provided by the MPA layer
    - Managing the iSCSI markers would mean that the RNIC have to be ULP-aware

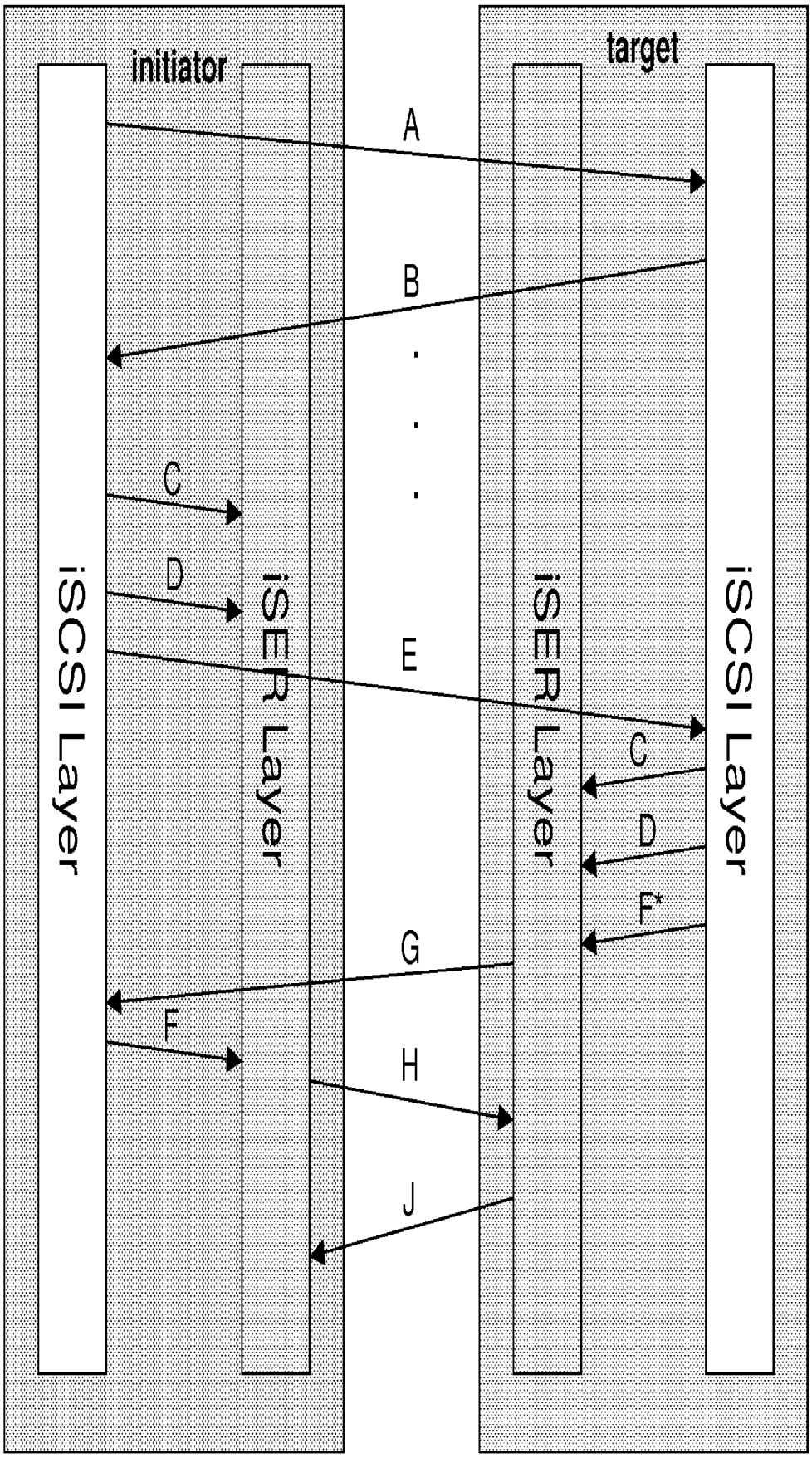
Connection Setup for iSER-assisted Mode at the Initiator

- Negotiated key values may be passed by the iSCSI layer to the iSER layer by invoking the Notice_Key_Values Operational Primitive
- Before sending the final Login Request, the iSCSI layer invokes the Allocate_Connection_Resources Operational Primitive to request the iSER layer to allocate the iWARP resources for the connection
- After the target returns the final Login Response, the iSCSI layer at the initiator invokes the Enable_Datamover Operational Primitive to request the iSER layer to transition into iSER-assisted mode
- The first message sent by the iSER layer at the initiator to the target is the iSER Hello Message

Connection Setup for iSER-assisted Mode at the Target

- Negotiated key values may be passed by the iSCSI layer to the iSER layer by invoking the Notice_Key_Values Operational Primitive
- Before sending the final Login Response, the iSCSI layer invokes the Allocate_Connection_Resources Operational Primitive to request the iSER layer to allocate the iWARP resources for the connection
- The iSCSI layer invokes the Enable_Datamover Operational Primitive to enable the iSER mode qualified with the final Login Response PDU
- The iSER layer sends the final Login Response PDU in byte stream mode and then transitions into iSER-assisted mode
- After receiving the iSER Hello Message from the initiator, the iSER layer at the target responds by sending the iSER HelloReply Message

Example of Successful iSER Connection Setup



A. SCSI Login Request PDU with RDMAExtensions=Yes

B. SCSI Login Response PDU with RDMAExtensions=Yes

C. Optional Notice_Key_Values to pass values of negotiated keys

D. Allocate_Connection_Resources to set up iWARP resources

E. SCSI Login Request PDU with T=1 and NSG=FullFeaturePhase

F. Enable_Datamover to go into iSER mode (* = send last iSCSI PDU in byte stream mode)

G. SCSI Login Response PDU in byte stream mode with T=1 and NSG=FullFeaturePhase

H. iWARP Send Message containing iSER Hello

J. iWARP Send Message containing iSER HelloReply

Unsuccessful Connection Setup for iSER-assisted Mode

- During the login phase, if both the initiator and the target fail to negotiate the RDMAExtensions key to “Yes”, then the connection continues with the semantics as defined in iSCSI

  s defined in iSCSI

  r reason it is not possible to enter iSER-assisted mode, the connection will

  will be terminated

    locating the iWARP resources, the iSCSI layer will terminate the connection

    ate the connection

    e iWARP resources, the iSCSI layer will send a Login Response with the cessful status and terminate the connection

    ate the connection

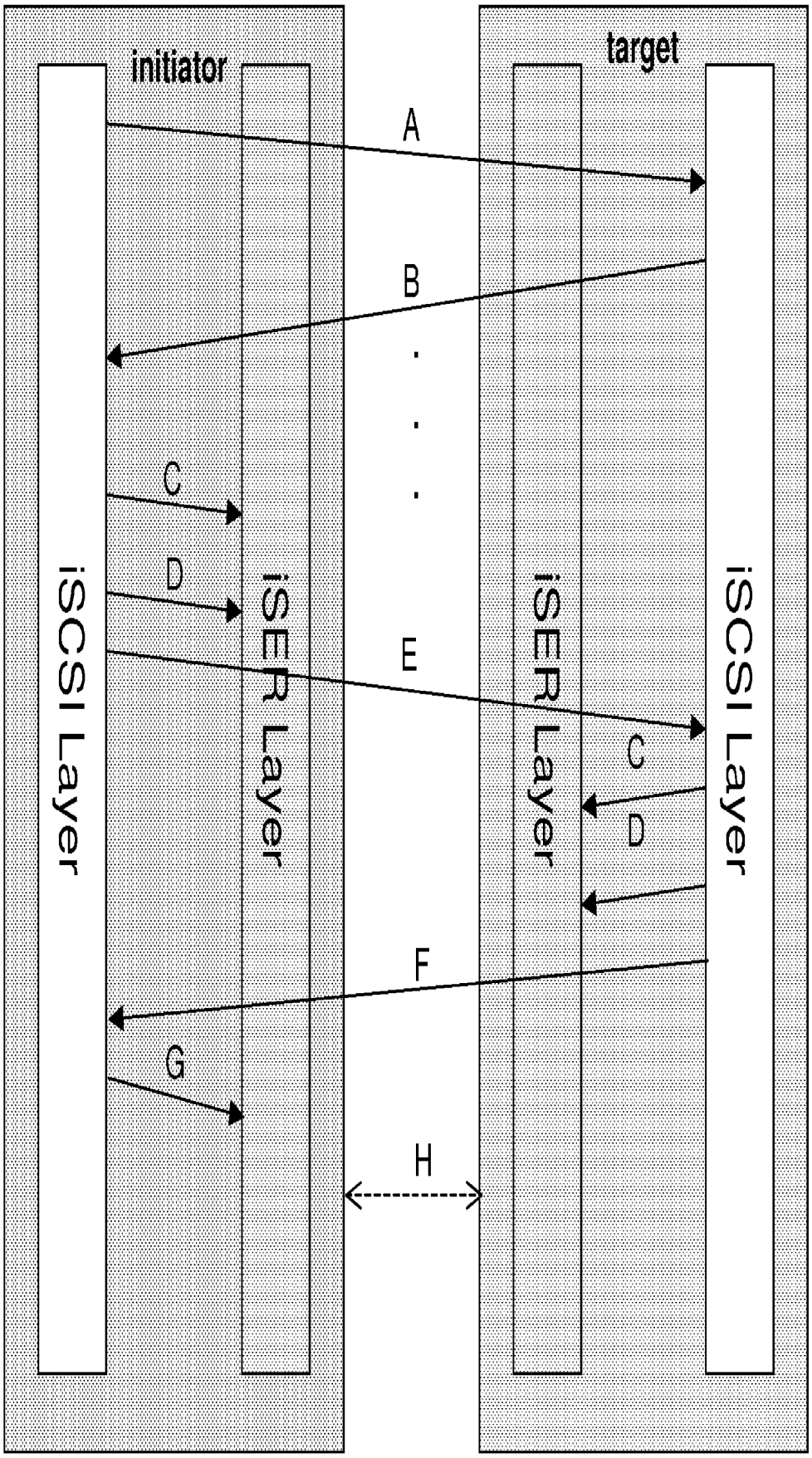
    in the iSER Hello Message is unacceptable to the target, the iSER layer will set the Reject flag in the iSER HelloReply Message and terminate the RDMAP

    e the RDMAP stream

      n_Terminate_Notify Operational Primitive after the RDMAP stream is terminated and all resources for the connection have been released

      have been released

Example of Unsuccessful iSER Connection Setup

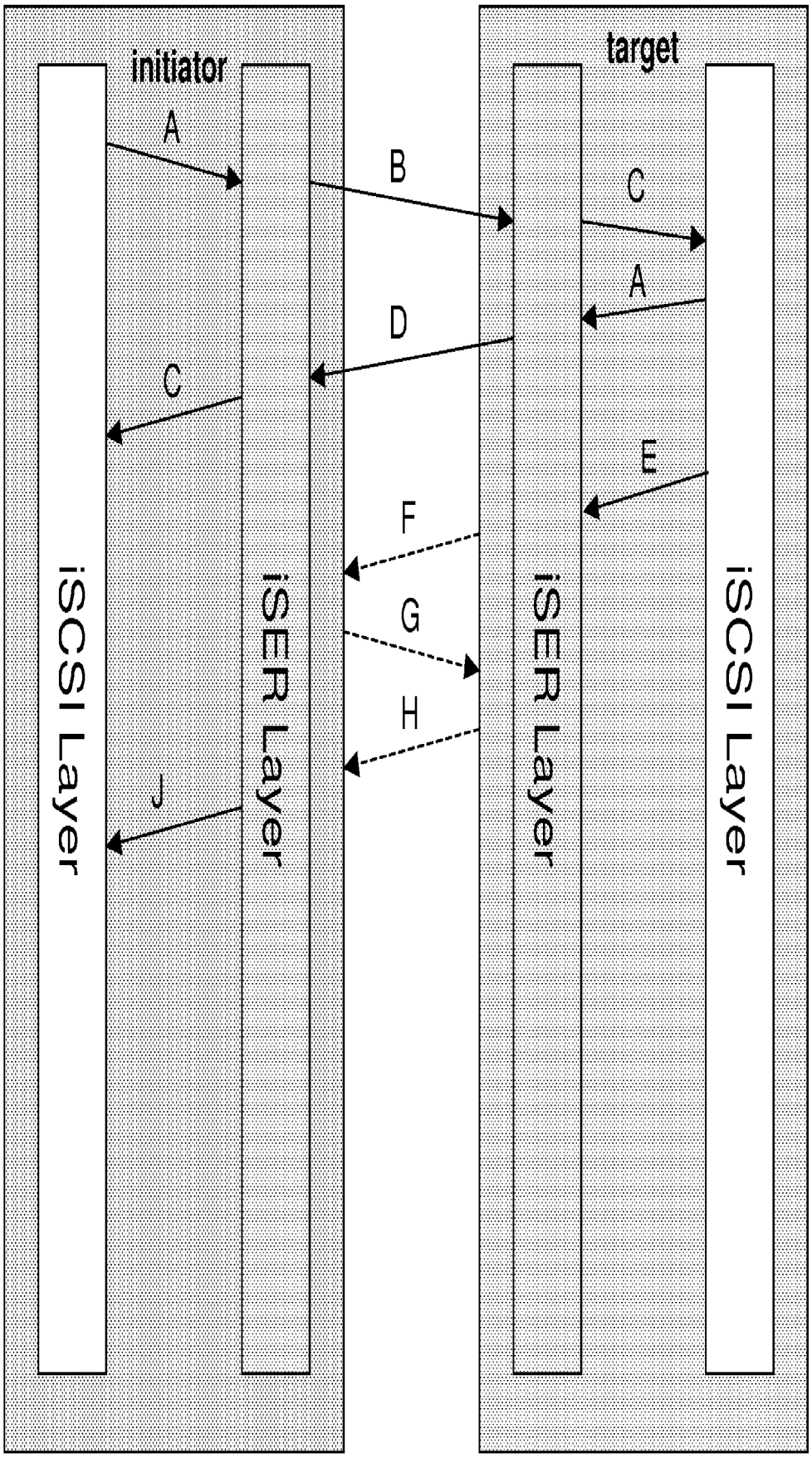


A. SCSI Login Request PDU with RDMAExtensions=Yes
B. SCSI Login Response PDU with RDMAExtensions=Yes
C. Optional Notice_Key_Values to pass values of negotiated keys
D. Allocate_Connection_Resources to set up iWARP resources
E. SCSI Login Request PDU with T=1 and NSG=FullFeaturePhase
F. SCSI Login Response PDU with unsuccessful status
G. Deallocate_Connection_Resources to release iWARP resources
H. TCP FIN exchanges to close the connection

Normal Connection Termination for iSER-assisted Mode

- The iSCSI layer at the initiator invokes the Send_Control Operational Primitive to request the iSER layer to send the Logout Request PDU
- The iSER layer at the target notifies the iSCSI layer of the Logout Request PDU by invoking the Control_Notify Operational Primitive
- The iSCSI layer at the target invokes the Send_Control Operational Primitive to request the iSER layer to send the Logout Response PDU
- The iSER layer at the initiator notifies the iSCSI layer of the Logout Response PDU by invoking the Control_Notify Operational Primitive
- After completing the iSCSI logout process, the iSCSI layer at the target invokes the Connection_Terminate Operational Primitive to request the iSER layer to terminate the RDMAP stream and release all resources for the connection
- After the TCP connection has been closed, the iSER layer at the initiator releases all resources for the connection and notifies the iSCSI layer by invoking the Connection_Terminate_Notify Operational Primitive

Example of Normal iSER Connection Termination



A. Send_Control to send SCSI PDU
B. iWARP Send Message containing SCSI Logout Request PDU
C. Control_Notify to report SCSI PDU received
D. iWARP Send Message containing SCSI Logout Response PDU
E. Connection_Terminate to close the connection and release iWARP resources
F. TCP FIN
G. TCP FIN + Ack
H. TCP Ack
J. Connection_Terminate_Notify to report that the connection is closed

Error Recovery

- All three ErrorRecoveryLevels as defined in iSCSI may be deployed in the iSER-assisted mode
  - The following considerations were made in order to support ErrorRecoveryLevels 1 and 2
- For ErrorRecoveryLevel 1
  - The iSCSI layer at the initiator should disable timeout-driven proactive SNACKs and timeout-driven PDU retransmissions since digest and sequence errors will not occur in the iSER-assisted mode
  - The PDU recovery realized via this ErrorRecoveryLevel will never be necessary since digests are not used and hence may be considered always supported
- For ErrorRecoveryLevel 2
  - When the iSCSI layer at the target accepts a reassignment request for a SCSI Read Command, it will invoke the Put_Data Operational Primitive to request the iSER layer to process the SCSI Data-in PDU if not all data is acknowledged
    - The iSCSI layer at the initiator will set ExpDataSN = 0 on Task Allegiance Reassignment to allow the target to request all unacknowledged data
  - When the iSCSI layer at the target accepts a reassignment request for a SCSI Write Command, it will invoke the Get_Data Operational Primitive to request the iSER layer to process the R2T PDU for any non-immediate unsolicited data and any solicited data that have not been received
    - Data previously designated as unsolicited will also be transferred using RDMA Read operations
- Note that iSCSI data acknowledgement support for ErrorRecoveryLevels 1 and 2 are described in slide 24

Summary

- The iSCSI Extensions for RDMA (iSER) is based on the Datamover Architecture. The iSER protocol allows the data movement and placement aspects of iSCSI to be offloaded to the iWARP protocol suite
  - Provides the option of using generic RNICs for iSCSI instead of dedicated iSCSI HBAs
    - Requires no iSCSI or iSER specific assists in the iWARP protocol suite or RNIC
  - Enables direct data placement of in-order or out-of-order SCSI data into pre-allocated SCSI buffers
    - Eliminates the data copy in the receive path to move the data to the final buffer
    - Eliminates unnecessary memory bandwidth consumption
    - Decreases reassembly buffer size requirements
    - Reduces CPU utilization
- iSER requires no changes to SCSI Architecture Model (SAM/SAM-2/SAM-3) and SCSI Command set standards
- iSER fully utilizes existing iSCSI infrastructure including but not limited to MIB, bootstrapping, negotiation, naming and discovery, and security
- iSER seeks to minimize impacts to existing iSCSI implementations in supporting the extensions
- For additional information, go to the RDMA Consortium website at www.rdmaconsortium.org

RDMAC Storage Subgroup Voting Record

Mike Ko

Updated as of 1/24/2009 3:58:00 PM

Note that some storage-related votes have been duplicated from the RDMAC Voting Record and meeting minutes including those from the Contributors WG and the Founders Group.

RWG-mtg-020626-dg.doc (F2F)

Microsoft Motions: RDMAC should explore a SCSI storage mapping to iWARP.
IBM Seconds
Motion passes by acclimation

Microsoft Motions: RDMAC shall use the iSCSI infrastructure with the possible exception of data transfer and login.
IBM Seconds
Motion passes by acclimation

Microsoft Motions: That RDMAC set the initial scope of the Storage sub-team to be "explore SCSI data transfer on iWARP using iSCSI infrastructure and login".
Adaptec Seconds
Motion passes by acclimation

RFG-mtg-020627.doc (RR)

Microsoft motions that the RDMAC enable the working group to setup a sub-team which is scoped to explore SCSI data transfer on iWARP using iSCSI infrastructure and login.
MS motioned, EMC seconded via e-mail.
results:

Adaptec	Yes
Broadcom:	Yes
Cisco	
EMC	Yes – via e-mail
HP	Yes
IBM	Yes
Intel	Yes
Microsoft	Yes
Network Appliance	Yes

Motion passes.

After the call Renato *noticed the operating procedures also state that: "The Working Group may, at its convenience, establish sub-groups to deal with individual topics or sections of a proposal or specification. The chairs of each sub-group will be appointed by the Working Group chairpersons." Renato suggest use the process we discussed during the call, because it provides a more fair distribution of co-chair appointments (i.e. entire founders group vs discretion of the Working Group co-chairs). Another alternative that would also be fair, would be to have the members of the contributors group select the co-chairs.*

RWG-mtg-020712-dg.doc

Motion to chose 2 of the three choices for storage sub-group co-chairs:

Adaptec	Jim & Hamel
Broadcom:	Jim & Mike
Cisco	
EMC	Jim & Mike
HP	Jim & Mike
IBM	Jim & Mike
Intel	Hamel & Mike
Microsoft	Jim & Mike
Network Appliance	Jim & MIke

Jim P: The two new co-chairs are Jim Wendt & Mike Ko.

Microsoft motions that electing subgroup co-chairs be on an individual basis and not a company basis.
Renato: IBM seconds.
Broadcom Abstains. Motion passes

RWG-020809.doc (ED)

Motion from email discussion: iSCSI 1.0 interoperability as represented by the following table is required of an endnode.
Adaptec seconds.

(from email – xxx – xx/xx/xx)

```
1.0 = iSCSI 1.0 support
iW = iWARP support

Capability  Capability        Resultant Negotiated
Initiator   Target            Data Transfer model after login

1.0         1.0               1.0
iW/1.0      1.0               1.0
```

```
1.0          iW/1.0            1.0
iW/1.0       iW/1.0            iW
```

Roll call:

Adaptec	Y
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Microsoft	Y
Network Appliance	Y

Motion passed unanimously.

RSTG-mtg-020812-jw.doc

HP Motion - "The Storage Subgroup requests a modification of the charter from the Founders to do a mapping function wire protocol and specification for iSCSI to iWARP and that the storage subgroup owns binding votes in these respective areas"

MS seconds motion

Roll call vote

Adaptec	Y
Broadcom:	Absent
EMC	Y
HP	Y
IBM	Y
Intel	Y
Microsoft	Y
Network Appliance	Absent

Motion passes

HP Motions "- The Storage Subgroup requests that the Working Group delegate investigation and definition of new verbs that are motivated by storage and that the Working Group delegate to the Storage Subgroup binding votes in this area"

IBM Seconds

Motion passes by Acclimation

HP Motions "The Storage Subgroup requests that the Working Group delegate investigation and definition of modifications to existing verbs that are motivated by storage and that the Working Group delegate to the Storage Subgroup binding votes in this area"

IBM Seconds

Motion passes by Acclimation

HP Motions "The Storage Subgroup requests that the Working Group delegate investigation and recommendation of the immediate data mechanism for the iWARP protocols to be completed 8/16"

IBM Seconds

Motion passes by Acclimation

RSTG-mtg-020814-jw.doc

EMC Motion - the Storage Subgroup finds that Immediate Data is NOT necessary

MS seconds

Intel - Friendly Amendment - the Storage Subgroup finds that Immediate Data is NOT necessary for SCSI mapping

MS seconds

Binding vote - **"The Storage Subgroup finds that Immediate Data is NOT necessary for SCSI mapping."**

Adaptec	Y
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Microsoft	Y
Network Appliance	Absent

Motion passes

RWG-mtg-020814-jw.doc

MS motions - all of a) and all of b) be accepted

<<<

JH - read Jim Wendt's email (8/12 - Request for agenda item - 8/14 Contributors WG meeting): as follows:

```
---------
The Storage Subgroup requests the following agenda item
be added for the 8/14 Contributors Working Group meeting:

* Storage Subgroup Delegations

  a) The Storage Subgroup requests that the Working Group
     delegate investigation and definition of new verbs that
     are motivated by storage and that the Working Group
     delegate to the Storage Subgroup binding votes in this area.

  b) The Storage Subgroup requests that the Working Group delegate
     investigation and definition of modifications to existing verbs
     that are motivated by storage and that the Working Group
     delegate to the Storage Subgroup binding votes in this area.

  c) The Storage Subgroup requests that the Working Group delegate
     investigation and recommendation of the immediate data mechanism
     for the iWARP protocols to be completed 8/16.

Thanks,
JimW
--------
```

JH - item c) was already done this morning in Storage subgroup call
>>>

IBM seconds

a) The Storage Subgroup requests that the Working Group delegate investigation and definition of new verbs that are motivated by storage and that the Working Group delegate to the Storage Subgroup binding votes in this area.

b) The Storage Subgroup requests that the Working Group delegate investigation and definition of modifications to existing verbs that are motivated by storage and that the Working Group delegate to the Storage Subgroup binding votes in this area.

Motion passes by Acclimation

HP Motions that - "The SCSI Write data transfer mechanism in the recommendation from the Storage Subgroup will be RDMA Reads"

Adaptec seconds

Binding roll call vote:

Adaptec	Y
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Microsoft	Y
Network Appliance	Y

Motion passes

RFG-mtg-020814.doc (RR)

The Founder WG charters the Storage Subgroup to do an iSCSI to iWARP mapping function and associated wire protocol. The storage subgroup owns binding votes in these respective areas and must coordinate with other standard bodies (e.g. ANSI T10 and IETF) for the SCSI encapsulation.

Intel motions to accept the text as captured in the sentence immediately above, Adaptec seconds:

Adaptec	Y
Broadcom:	Y
Cisco	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Microsoft	Y
Network Appliance	Y

(motion passes)

RSTG-mtg-020819-mk.doc

Microsoft motions that – "Create a semantics in the send message that encapsulates the SCSI response that will provide an invalidate STag capability"
IBM seconds
Amended motion - "**Create a semantics in the Send and SendSE messages only that encapsulates the SCSI response that will provide an invalidate STag capability**"
Object – NetApp
Abstain – None
Motion passes 7 to 1

RWG-020819.doc (ED)

HP IBM motions that the Send with "invalidate STag message" and Send with SE and "invalidate STag message" be added to the RDMA protocol. When that message is received the data sink will invalidate the STag contained in the message.
Microsoft seconds.
Motion passes by acclamation with one abstention and one absent.

RWG-mtg-020821-jw.doc

MS Motions - "The STag invalidate feature be implemented as an extension of the existing DDP control field in untagged messages only from one to five bytes and that the DDP control field be valid in every segment of the DDP message"
IBM seconds
Motion passes by Acclimation

HP Motions - "We will not extend the Reserved ULP data field in tagged messages for DDP"
IBM seconds
Amended Motion - "We will not extend the Reserved ULP data field in tagged messages for DDP beyond one byte"
IBM accepts friendly amendment
Motion passes by Acclimation

IBM Motions - "The Send with Invalidate Stag message and Send with SE Invalidate Stag message be added to the RDMA protocol. When either of those messages are received, the data sink will invalidate the Stag in the message."
MS Seconds

Amended Motion - "The Send with Invalidate Stag message and Send with SE Invalidate Stag message be added to the RDMAP protocol. When either of those messages is received at the Data Sink, the Data Sink will invalidate the Stag contained in the message if that Stag is associated with the RDMA Stream."

IBM accepts friendly amendment
MS accept friendly amendment
Motion passes by acclimation

RSTG-mtg-F2F-020905-jw1.doc

IBM motions "we accept the schedule that we worked out on 9/4 F2F"

Dates copied from slide displayed by MK

- v0.1 – 9/12/02
- v0.5 – 10/22/02
- v0.7 - 12/13/02
- v0.8 – 1/17/03
- v0.9 – 2/14/03
- v0.95 – 3/7/03
- v1.0 – 4/14/03

Adaptec seconds

Microsoft friendly amendment "we accept the iPER (iSCSI Protocol Extensions for RDMA) and DI (Datamover Interface) schedule that we worked out on 9/4 F2F"

IBM accepts friendly amendment
Adaptec accepts friendly amendment
Binding vote:

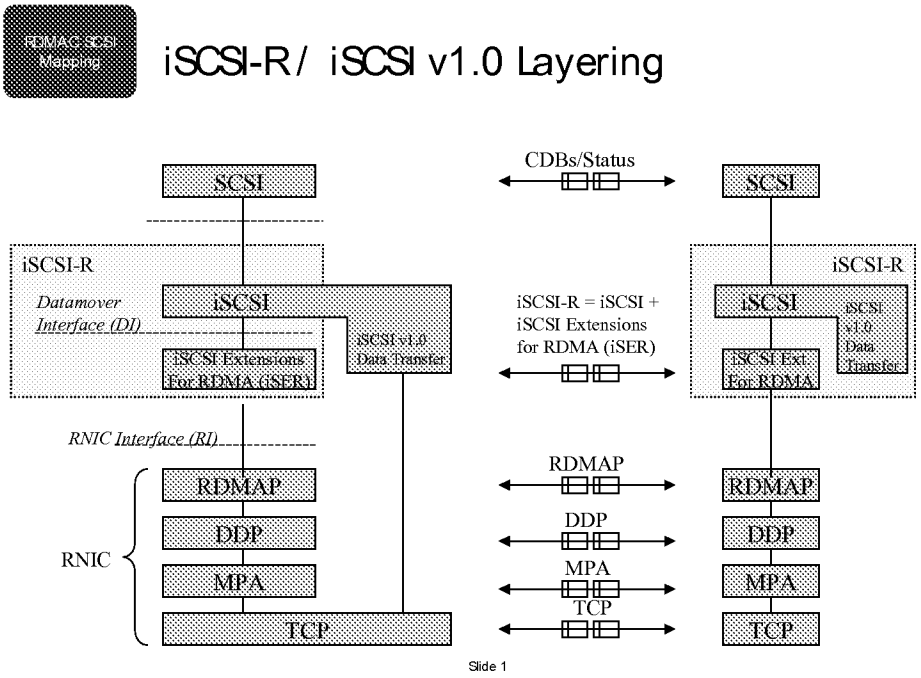
Adaptec	Y
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Microsoft	Y
Network Appliance	Y

Motion passes

RSTG-mtg-0209091.doc (ED)

HP motions to adopt the iSCSI-R layering slide

(from slide – iSCSI-R-Layering-v1.ppt – 9/10/02 - Jim Wendt)



with the following four changes:

- TCP Data Transfer changes to iSCSI V1.0 data transfer
- iSCSI protocol extensions for RDMA (iPER) changes to iSCSI extensions for RDMA (iSER)
- Remove the TOE I/F and the associated dashed line
- Change the RDMAP Ops or Verbs to RNIC Interface (RI)

and two additional changes:

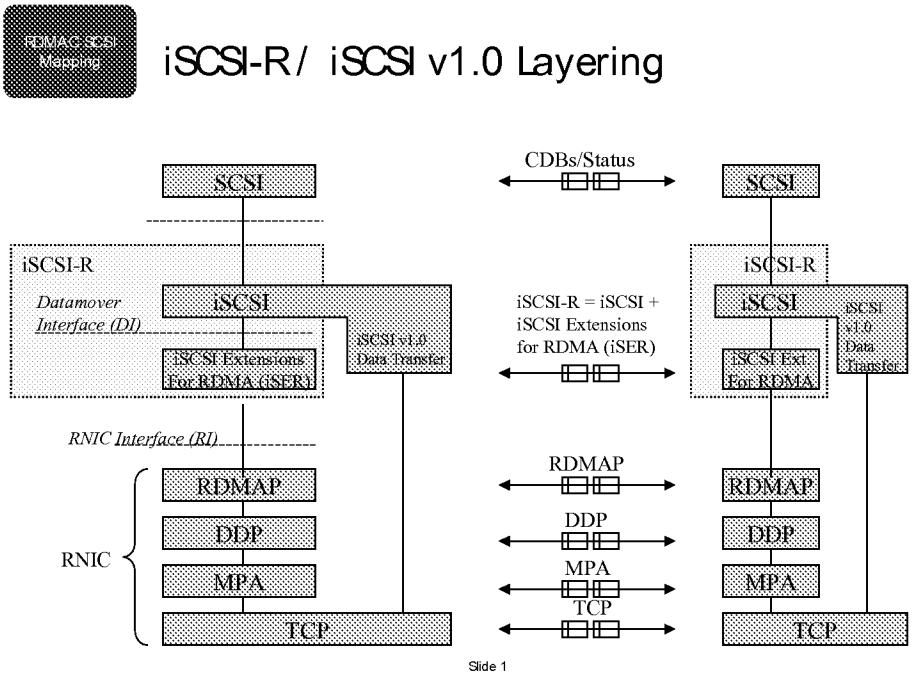
- iSCSI-R V1.0 data transfer block to the RHS of the diagram. (To be symmetric).
- Add iSCSI V1.0 to the title of the slide.

Broadcom seconds.

Motion passes by acclamation.

The updated slide is as follows:

(slide from – RDMAC iSCSI-R-layering-v2.ppt – 9/10/02 - John Hufferd)



IBM motions that we accept the first bullet as a requirement.

MikeK: Changed from no changes to the iSCSI Architecture SAM to no major changes.

- iSCSI-R must not require major changes to SCSI Architectural Model (SAM/SAM-2/SAM-3) and SCSI Primary Commands (SPC-3)

Intel seconds with the friendly amendment to spell out SPC, SPC-2, SPC-3 instead of just SPC-3.

IBM accepts the friendly amendment.

Motion passes by acclamation.

IBM motions that we accept the second bullet as a requirement.

Second bullet:

- iSCSI-R must utilize existing iSCSI infrastructure including but not limited to MIB, bootstrapping, iSNS, naming & discovery, security, etc.

Intel seconds.

Motion passes by acclamation.

IBM motions to that we accept the third bullet, modified as above, as a requirement.

Third bullet:

- iSCSI-R must not be required to interoperate with existing iSCSI v1.0 implementations

Changed wording to read:

- iSER is not required to interoperate with iSCSI v1.0 implementations in full feature phase.

Intel seconds.

Motion passes by acclamation.

IBM motions to accept the entire session management slide deck.
EMC seconds
Motion passes by acclamation.
(slides from – session_management.ppt - 9/7/02 - Mike Ko)

Session Management

- No change in iSCSI's definition that an I_T nexus maps to a session
- No changes to mechanisms described in iSCSI for:
  - Connection reinstatement
  - Session reinstatement
  - Session continuation
  - Session closure
  - Session failure
- Votable: iSCSI-R MUST use the mechanisms described in iSCSI for connection reinstatement, session reinstatement, session continuation, session closure, and session failure

Slide 1

RSTG-mtg-0209011.doc (ED)

HP motions that we accept the following requirement: iSCSI-R must not require any additional mechanisms over those provided by the MPA/DDP/RDMAP wire protocols, RNIC hardware and RNIC Interface (RI).
Broadcom second.
Motion passes by acclamation.

HP motions that the DI spec for 0.1 outline contained in the 3 slides 11, 12 and 13 be accepted.
Intel seconds.
Motion passes by acclamation
(slides from – Datamover_Interactions_v0.20.ppt – 9/9/02 - Mallikarjun Chadalapaka)

RDMAC iSCSI Mapping

DI spec outline

Following is the proposed DI spec outline.

1. Status of this Memo
2. Abstract
3. Introduction
4. Motivation
5. Terminology
   - 5.1 iSCSI PDU types
   - 5.2 Data Descriptor
   - 5.3 Connection_Handle
6. Datamover layer
7. Operational Primitives required by iSCSI
   - 7.1 Send_Control
   - 7.2 Send_Data
   - 7.3 Retrieve_Data

V0.20 Slide 11 September 09, 2002

RDMAC iSCSI Mapping

DI spec outline

8. Operational Primitives provided by iSCSI
   - 8.1 Control_Notify
   - 8.2 Connection_Terminate_Notify
   - 8.3 Data_Completion_Notify
   - 8.4 Data_ACK_Notify

9. Architectural layering of iSCSI and Datamover layers
10. Modeling description for the interactions
    - 10.1 Overview
    - 10.2 Interactions for handling asynchronous notifications
      - 10.2.1 Connection termination
      - 10.2.2 Data transfer completion
        - 10.2.2.1 Completion of a requested SCSI Data transfer
      - 10.2.3 Data acknowledgement
    - 10.3 Interactions for sending an iSCSI PDU

      Subsections for each iSCSI v1.0 opcode.

V0.20 Slide 12 September 09, 2002

DI spec outline

10.4 Interactions for receiving an iSCSI PDU
Subsections for each iSCSI v1.0 opcode.
11. Security Considerations
12. IANA Considerations
13. References and Bibliography
13.1 Normative References
13.2 Informative References
14. Authors' Addresses
15. Acknowledgements
Appendix A. Datamover considerations

Votable: HP motions that the DI spec outline contained in these 3 slides be accepted.

V0.20 Slide 13 September 09, 2002

RSTG-mtg-020916-jc.doc

HP motions to make Agilent voting member of storage subgroup
IBM seconds
Motion passes by acclamation

IBM motions that the iSER outline as described be accepted
HP seconds
Motion passes by acclamation
(slides from – iser_outline_v01.ppt - 9/11/02 – Mike Ko)

RDMAC iSCSI Mapping

iSER Specification Outline

1. Definitions and acronyms
2. Overview
   Contains the motivation and architectural goals, and provides overviews on iSCSI compatibility, iSCSI-R layering, iSCSI session/ stream mapping, iSCSI-R data movement request/ response scenarios, etc.
3. Login and full feature phase negotiation
   Describes the session management mechanisms, session and connection initialization, steering mode key negotiation, mixed mode sessions (if any), etc.
4. Lower layer protocol requirements
   Describes the RDMAP requirements
5. Upper layer interface requirements
   Refers to the Datamover Interface
6. iSER error handling and recovery
   Describes the impact of MPA CRC on iSCSI, support of iSCSI digests, ErrorRecoveryLevel supported, allegiance reassignment, etc.

Slide 2

RDMAC iSCSI Mapping

Specification Outline (cont.)

7. Security considerations
8. iSER PDU formats
   Describes the handling of iSCSI control and data PDUs between the Datamover Interface and the RDMAP layer
   Describes the formats of the control and data packets
   Describes STag management, supported PDUs (e.g., SNACK)
9. Login operational keys
   New login parameter requirements for iWARP
10. IANA considerations

- Appendix
  - Describes an iWARP PDU containing the iSCSI-R packet.
  - Describes an iSCSI transaction example (e.g., how a SCSI Read/ Write is handled by the different layers from the local peer to the remote peer)
  - RNIC spanning

Votable: IBM motions that the iSER spec version 0.1 outline contained in these 2 slides be accepted.

Slide 3

HP motions the DI design slide with the two changes described today be accepted (add new bullet #3, remove 'iWARP' from bullet #4)

- new bullet 3 – "The DI spec enables implementing the datamover layer either in hardware or in software"
- remove "iWARP" from (new) bullet #4

IBM seconds

Motion passes by acclamation

(slide from – Datamover_Interactions_v0.20.ppt – 9/9/02 - Mallikarjun Chadalapaka)

Design ideas

- An abstract procedural interface definition of iSCSI layer's interactions with a Datamover layer below – i.e. this models the interactions between the logical "bottom" interface of iSCSI and the logical "top" interface of a Datamover.
- The DI spec guides the SCSI mapping wire protocol (iSER) by defining the iSCSI knowledge iSER may utilize in its protocol definition (for ex., the DI spec completely contains the notion of "iSCSI session" to the iSCSI layer).
- Not a wire protocol spec, only a model for interactions between iSCSI and Datamover layers operating within an iSCSI/ iWARP endnode.
- The DI spec by design seeks to model the iSCSI-Datamover interactions in a way that the modeling is independent of the specifics of either a particular iSCSI revision, or a particular instantiation of a Datamover layer.
- Relies on a defined set of Operational Primitives (could be seen as entrypoint definitions in implementation terms) provided by each layer to the other to carry out the request-response interactions.

Votable: HP motions that the DI design ideas contained in this slide be accepted.

V0.20 Slide 2 September 09, 2002

RSTG-mtg-020918-pat.doc

HP - Moved that the operation requirements of a Datamover layer listed in slide 4 of Data_Mover_Interactions_v0.21 (Sept 18) be accepted.

Broadcom - seconds

passed by acclamation

Datamover layer?

- The following are proposed to be the operational requirements to be considered a "Datamover layer" per the DI spec.
  - guarantees that all the necessary data transfers take place when the local iSCSI layer requests transmitting a command (in order to complete a SCSI command, for an initiator),or sending/receiving an iSCSI data sequence (in order to complete part of a SCSI command, for a target).
  - transports an iSCSI control-type PDU to the peer Datamover layer when requested so by the local iSCSI layer.
  - provides a notification and delivery to the iSCSI layer upon arrival of an iSCSI control-type PDU.
  - provides an end-to-end data acknowledgement of iSCSI read data operations, when requested.
  - provides an asynchronous notification of data transfer operations upon completion.
  - places the SCSI data into the data buffers or picks up the SCSI data for transmission out of the data buffers that the iSCSI layer had requested be used for a SCSI I/O.
  - guarantees an error-free, reliable, in-order transport mechanism over IP fabrics in performing the data transfer, and asynchronously notifies the iSCSI layer upon iSCSI connection termination.

Votable: HP motions that the operational requirements of a Datamover layer that are listed in this slide be accepted.

V0.21 Slide 4 September 18, 2002

HP - Moved that iSCSI PDU class taxonomy contained in Slide 6 of Data_Mover_Interactions_v0.21 (Sept 18) be accepted.

Intel - seconds

passed by acclamation

iSCSI PDU classes

- Crucial difference between
  1. iSCSI PDUs that are transported intact to the peer iSCSI layer ("iSCSI control-type PDUs"), and
  2. iSCSI PDUs that trigger iSCSI-transparent RDMA transactions to effect data transfer ("iSCSI data-type PDUs"), without involvement from the peer iSCSI layer.
- Datamover layer is assumed to transport all iSCSI control-type PDUs using the same mechanism, such as the Send Message, and its peer Datamover layer reliably delivers them to the remote iSCSI layer.
- In case of iSCSI data-type PDUs, Datamover layer carries out the requested data transfer to the remote iSCSI node. However, the PDUs are *not delivered* to the remote iSCSI layer. There are only three data-type PDUs –
  a) R2T PDU
  b) iSCSI Data-in PDU
  c) iSCSI Data-out PDU (solicited only, unsolicited Data-out PDUs are control-type PDUs)
- All other iSCSI PDUs are termed as iSCSI control-type PDUs for this discussion.

Votable: HP motions that the iSCSI PDU class taxonomy contained in this slide be accepted.

V0.21 Slide 6 September 18, 2002

HP - Moved that the model for PDU interactions and connections described in slide 8 of Data_Mover_Interactions_v0.21 (Sept 18) be accepted except send_data and retrieve_data will be changed to put_data and get_data.
Intel - seconds
passed by acclamation

Modeling PDU interactions & Terminate

- All iSCSI control-type PDU exchanges can be modeled using the Send_Control and Control_Notify pair of Operational Primitives. This applies to the following iSCSI PDU opcodes in iSCSI v1.0.
  - SCSI Command/ Response, Task Mangement Function Request/ Response
  - Text Request/ Response, Login Request/ Response, Logout Command/ Response,
  - Asynchronous Message, Reject, NOP-In, NOP-Out, SNACK
- All iSCSI data-type PDU exchanges (actually it only results in transfer of data referenced by the PDU, not the PDU per se) can be modeled using
  a) Put_Data & Data_Completion_Notify pair of Operational Primitives (for outbound data transfer), **or**
  b) Get_Data & Data_Completion_Notify pair of Operational Primitives (for inbound data transfer)
- A Datamover layer notifies the iSCSI layer upon iSCSI connection termination via the Connection_Terminate_Notify Operational Primitive. The iSCSI layer is expected to process it as defined in the iSCSI spec.

Votable: HP motions that the model for PDU exchange interactions and connection termination described in this slide be accepted.

V0.21 Slide 8 September 18, 2002

HP _ Moved that the model for iSCSI data acknowlegements described on slide 10 of Data_Mover_Interactions_v0.21 (Sept 18) be accepted.
Intel - seconds
passed by acclamation

Modeling iSCSI Data acknowledgements

- The iSCSI spec defines an iSCSI Data Acknowledgement request (that can only originate from a target iSCSI layer) as an iSCSI Data-in PDU with the A-bit set. Such an iSCSI Data-in PDU triggers two eventual notifications in this proposed model.

  a) The first notification is for the completion of the outbound data transfer that is defined by the Data-in PDU and requested via the Put_Data Operational Primitive (the notification done via the Data_Completion_Notify Operational Primitive).
  b) The second notification is for a Datamover-managed data acknowledgement response arrival (done via the Data_ACK_Notify Operational Primitive).

Votable: HP motions that the model for iSCSI data acknowledgements described in this slide be accepted.

V0.21 Slide 10 September 18, 2002

RWG-mtg-020918-emd.doc

IBM motions that for an iSCSI command the initial TO field must start at zero.

Microsoft seconds.

MC: Not for an iSCSI command, it is iSER header carrying an iSCSI command assumes that the initial TO field is zero.

Microsoft accepts the

Motion passes by acclamation.

RSTG-mtg-020923-ue.doc

HP motions that "there shall be an iSER header"

Second: Intel

PASSES by acclamation

IBM motions that "iSER header shall be located between the iWARP header and the iSCSI PDU".

Intel - friendly amendment:

IBM motions that "for all iSCSI "control type" PDU, an iSER header is positioned following the iWARP header"

Intel seconds.

Passes by acclamation

HP motions: "There shall be a control field in the iSER header"
BROADCOM seconds.
Passes by acclamation

RSTG-mtg-020925-tc.doc

HP moves that the iSER header be fixed in size.
Intel seconds.
passes by acclamation.

HP moves that the iSER header contains a Write Stag followed by a Read Stag field and two bits in the control field tha indicates the validity of those two stags.
Intel seconds.
HP accepts friendly amenment (reflected above).
passes by acclamation.

Agilent moves that there be a version in the iSER header control field.
IBM seconds.
HP friendly amendment to put it in the control field.(reflected above)
Agilent - either way is OK. I accept that.
passes by acclamation.

IBM moves For iSCSI connections, iWARP mode is negotiated during the iSCSI login exchanges after TCP connection establishment.
Intel seconds.
Intel friendly amendment: reflected above.
Accepted by IBM and Intel.
passes by acclamation.

RSTG-mtg-020930-jw.doc

IBM motions (updated with amendments) – An RDMA mode key of the form "RDMAExtensions=<boolean-value>" with result function of AND available to both the initiator and the target only during iSCSI log-in with the default being No is used to determine iSER support for the iSCSI peers"
EMC seconds
Microsoft friendly amendment – use "iSER support" not "iWARP support" (reflected above)
IBM accepts friendly amendment
EMC accepts
MC – default should be "No" (reflected above)
MC – friendly amendment – change to call it "RDMA mode key" (reflected above)
EMC accepts

Motion passes by acclimation

RSTG-mtg-021002-mk.doc

IBM Motions - The same steering mode is not required for all connections of the same session.
EMC Seconds
Binding vote :

Adaptec	Absent
Agilent	N
Broadcom:	N
EMC	N
HP	N
IBM	Y
Intel	N
Microsoft	N
Network Appliance	Abstain

Motion fails

IBM motions - The same steering mode is required for all connections of the same session.
EMC seconds
Further discussion - none
Vote against – none
Abstain – IBM
Motion pass by acclimation

RSTG-mtg-021007.doc (ED)

Intel motions: The Read STag used in a bi-directional command is invalidated with a Send with Invalidate and Solicited Event message. If a Write STag is present in a bi-directional command, it is invalidated by the initiator using local means.
IBM seconds.
Motion passes by acclamation.

RSTG-mtg-021010.doc (TT/JP - 10/9 Storage F2F)

IBM moves the following: **iSER shall support unsolicited data transfer (both immediate and non-immediate) using only RDMAP Send message type. [Roll call vote]**

Microsoft seconds.

Binding vote:

Agilent	Y
Adaptec	Y
Broadcom:	Maybe Yes
EMC	Y
HP	Y
IBM	Y
Intel	Y
Microsoft	Y
Network Appliance	Y

Motion passes unanimously.

IBM moves the following: **An iSCSI command and its associated immediate data, if any, shall be transferred in a single RDMAP Send message.**

Agilent seconds.

Friendly amendment: add "segment", delete immediate.

An iSCSI command and its associated iSCSI data segment (e.g. immediate data), if any, shall be transferred in a single RDMAP Send message.

IBM and Agilent accept amendment.

Motion passes by acclamation.

IBM moves: **Each iSCSI Unsolicited DataOut pdu shall be transferred through one RDMAP Send message type.**

Intel seconds.

Motion passes by acclamation.

IBM moves: **The number and size of SGL elements for the iSER receive queue is implementation dependent.**

Broadcom seconds.

Motion passes by acclamation.

HP moves: **ISER shall support a new iSCSI login key that requires the Initiator to fill all non-last unsolicited data to a negotiated, fixed size (including immediate data and DATAOUT PDU)**

Agilent seconds.

Friendly amendments:

ISER shall support a new iSCSI login key that requires the Initiator to fill all non-last unsolicited data PDUs in an iSCSI data sequence to a negotiated, fixed size (including immediate data and DATAOUT PDUs). The key shall be named FullSendDataSegmentLength.

Friendly amendment(s) accepted

Motion passes by acclamation.

Intel moves: **We approve the "Solicited Data Handling in iSER v0.7" slide deck as modified in the meeting.**

HP seconds.

Motion passes by acclamation.

Solicited Data Handling in iSER v0.7

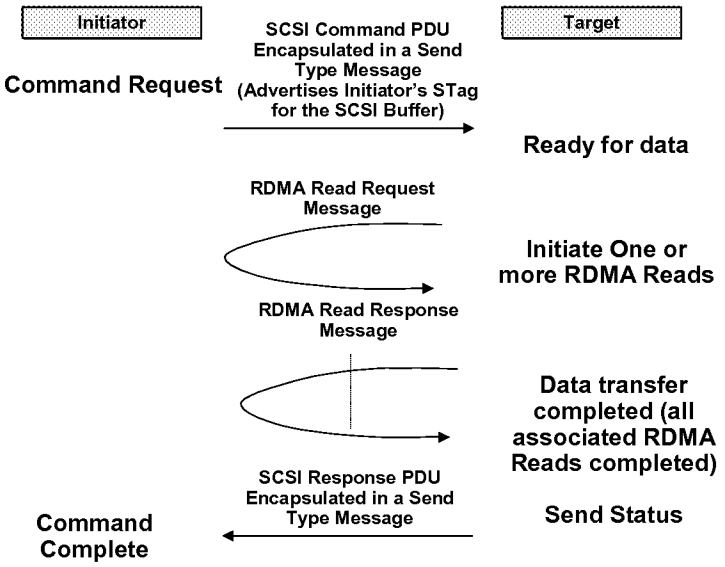
Hemal Shah
Mike Ko
Uri Elzur
October 9, 2002

Solicited Data Handling

- iSCSI-R Target initiates solicited data transfer for both SCSI write and SCSI reads
- iSER layer on the initiator is not involved in solicited data transfer
- iSER layer on the initiator is responsible for advertising the STag corresponding to a SCSI buffer used for a SCSI write or a SCSI read
- Solicited data for SCSI writes are handled using RDMA Read operation(s)
  - Each R2T PDU is transformed into a RDMA Read operation
  - No Data-out PDUs are generated at the initiator
- SCSI reads are handled using RDMA Write operation(s)
  - All SCSI read data is implicitly solicited by command
  - Each Data-In PDU is transformed into a RDMA Write operation

iSCSI-R Mapping Slide 2

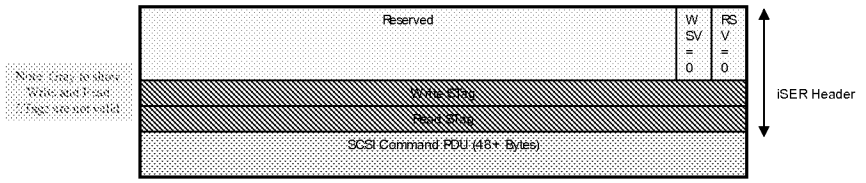
Solicited Data Transfer for a SCSI Write



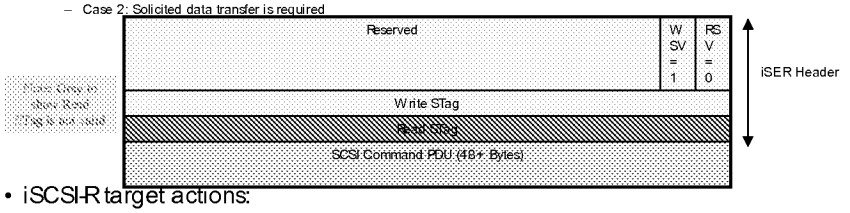
iSCSI-R Mapping Slide 3

Handling of SCSI Command PDU for SCSI Write

- iSCSI-R initiator actions:
  - iSER layer at the initiator receives SCSI Command PDU in Send_Control primitive from the iSCSI layer
  - iSER layer at the initiator associates an STag to the SCSI buffer used for the solicited data and places it in the Write STag field of the iSER header
    - e.g. iSER uses a pre-allocated STag and uses Send and Register verb to combine registration with Send Type operation
  - SCSI command PDU containing the write command is embedded in a Send Type Message (Message Payload as shown in the the following figure)
    - Case 1: No solicited data transfer is required



    - Case 2: Solicited data transfer is required



- iSCSI-R target actions:
  - Upon receiving the Send Type Message containing SCSI command PDU, iSER layer on the target notifies iSCSI layer using Control_Notify primitive
    - iSER layer makes an association of Write STag and ITT on this connection
    - iSER layer only passes SCSI command PDU to the iSCSI layer

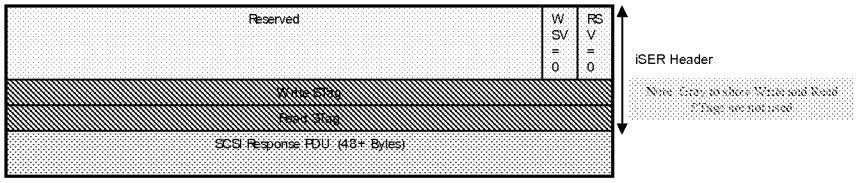
iSCSI-R Mapping Slide 4

Handling of R2T for SCSI Write

- iSCSI-R target actions:
  - Target uses Get_Data primitive one or more times to complete data transfer
  - The following steps are performed for each Get_Data operation:
    - For solicited data, when a buffer becomes available, the target iSCSI layer uses Get_Data primitive (containing R2T PDU) to notify iSER layer
    - After receiving Get_Data primitive, the target iSER layer, issues an RDMA Read request (combined with local register and deregister)
      - Write STag corresponding to the ITT in R2T PDU is used as Data Source STag
      - Buffer Offset field of R2T PDU is used as Data Source TO
      - Desired Data Transfer Length field of R2T PDU is used as Size for RDMA Read operation
      - iSER layer associates R2TSN with this RDMA Read operation
    - Upon completion of every RDMA read operation, iSER layer notifies the iSCSI layer using Data_Completion_Notify primitive that includes ITT and R2TSN
- iSCSI-R Initiator Actions:
  - iSER and iSCSI layers at the initiator are not involved in the data transfer associated with the R2T
    - RDMA Read Response Message carries solicited SCSI data

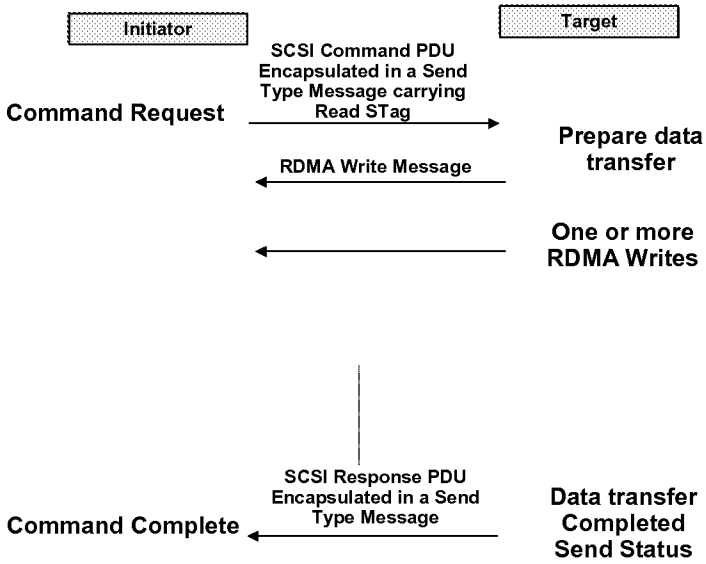
Handling of SCSI Response PDU for SCSI Write

- iSCSI-R Target Actions:
  - iSER layer at the target receives SCSI Response PDU in Send_Control primitive from the iSCSI layer
  - SCSI Response PDU from the target is embedded in a Send Type Message (Message Payload as shown in the the following figures)
    - iSER Header followed by SCSI Response PDU



- iSCSI-R initiator actions:
  - Upon receiving the Send Type Message containing SCSI Response PDU, iSER layer on the initiator notifies iSCSI layer using Control_Notify primitive

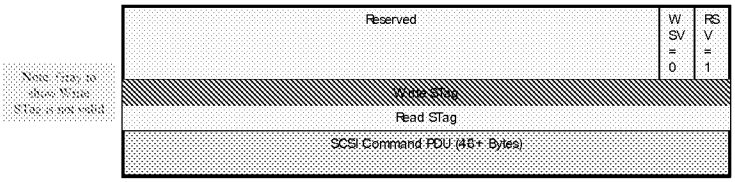
SCSI Read Data Transfer



iSCSI-R Mapping Slide 7

Handling of SCSI Command PDU for SCSI Read

- iSCSI-R Initiator Actions:
  - iSER layer at the initiator receives SCSI Command PDU in Send_Control primitive from the iSCSI layer
  - iSER layer at the initiator associates an STag to the SCSI buffer used for the solicited data and places it in the Read STag field of the iSER header
    - e.g. iSER uses a pre-allocated STag and uses Send and Register verb to combine registration with Send Type operation
  - SCSI command PDU from the initiator containing the read command is embedded in a Send Type Message (Message Payload as shown in the the following figure)



- iSCSI-R Target Actions:
  - Upon receiving the Send Type Message containing SCSI command PDU, iSER layer on the target notifies iSCSI layer using Control_Notify primitive
    - iSER layer makes an association of Read STag and ITT on this connection
    - iSER layer only passes SCSI command PDU to the iSCSI layer

iSCSI-R Mapping Slide 8

Handling of RDMA Write for SCSI Read

- iSCSI-R Target Actions:
  - For solicited data, when the data becomes available, the target iSCSI layer uses Put_Data primitive to notify iSER layer
  - Target performs one or more Put_Data operations to complete the data transfer
  - For each Put_Data:
    - After receiving Put_Data primitive, the target iSER layer, issues an RDMA Write command
      - Read STag received from the initiator in SCSI command PDU is used as the STag for RDMA Write
      - Buffer Offset field of SCSI Data-In PDU is used as TO
      - DataSegmentLength field of SCSI Data-In PDU is used as Size for RDMA Write operation
      - iSER layer associates DataSN with this RDMA Write operation
    - Upon completion of every RDMA write operation, iSER layer notifies the iSCSI layer using Data_Completion_Notify primitive that includes ITT and DataSN
- iSCSI-R Initiator Actions:
  - iSER and iSCSI layers at the initiator are not involved in the data transfer
    - RDMA Write Message carries solicited SCSI data

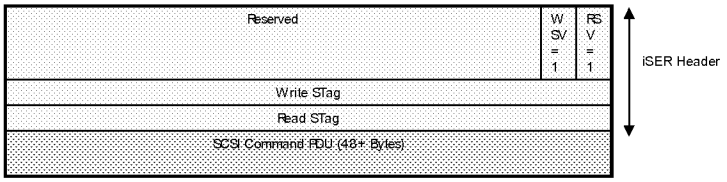
Handling of SCSI Response PDU for SCSI Read

- iSCSI-R Target Actions:
  - iSER layer at the target receives SCSI Response PDU in Send_Control primitive from the iSCSI layer
  - SCSI Response PDU from the target is embedded in a Send Type Message (Message Payload as shown in the the following figures)
    - iSER Header followed by SCSI Response PDU

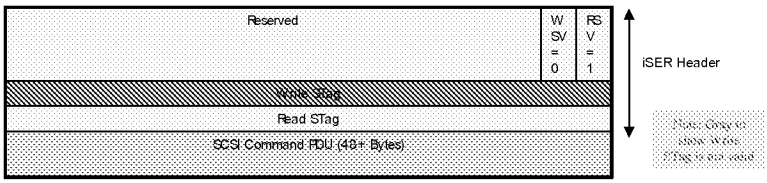
- iSCSI-R initiator actions:
  - Upon receiving the Send Type Message containing SCSI Response PDU, iSER layer on the initiator notifies iSCSI layer using Control_Notify primitive

Bi-directional Commands

- For Bi-directional commands (SCSI read and SCSI Write), the solicited data transfer is handled in a similar fashion
  - For SCSI read part, same as described on Slide 9 and for SCSI write part, same as described on Slide 5
- Message Payload of Send Type Message carrying SCSI command PDU
  - Case 1: SCSI Write requires solicited data transfer



  - Case 2: SCSI Write does not require solicited data transfer



- SCSI Response is handled in a similar fashion as described on slide 6 or slide 10

iSCSI-R Mapping Slide 11

Type of Messages and Invalidation of STags

- Initiator shall use Send with SE for carrying SCSI command PDU
- If there were no STags advertised (WSV=RSV=0 in iSER header) in the Send with SE Message containing SCSI command PDU
  - Target shall use Send with SE Message for carrying the SCSI Response PDU
- If Write STag and/ or Read STag were advertised in the Send with SE Message containing SCSI command PDU
  - Target shall use Send with Invalidate and SE Message for carrying the SCSI Response PDU
- If Read STag was advertised (RSV=1 in iSER Header) in the Send with SE Message carrying SCSI command PDU, then Send with Inv. and SE carrying SCSI Response PDU shall have Read STag in the RsvdULP field of the DDP Header
- If Write STag was advertised and Read STag was not advertised (WSV=1, RSV=0 in iSER Header) in the Send with SE Message carrying SCSI command PDU, then Send with Inv. and SE carrying SCSI Response PDU shall have Write STag in the RsvdULP field of the DDP Header
- Receive completion event corresponding to Send with Inv. and SE provided to iSER shall include the value of the invalidated STag

iSCSI-R Mapping Slide 12

Type of Messages and Invalidation of STags

- Initiator shall use Send with SE for carrying SCSI command PDU
- If there were no STags advertised (WSV=RSV=0 in iSER header) in the Send with SE Message containing SCSI command PDU
  - Target shall use Send with SE Message for carrying the SCSI Response PDU
- If Write STag and/ or Read STag were advertised in the Send with SE Message containing SCSI command PDU
  - Target shall use Send with Invalidate and SE Message for carrying the SCSI Response PDU
- If Read STag was advertised (RSV=1 in iSER Header) in the Send with SE Message carrying SCSI command PDU, then Send with Inv. and SE carrying SCSI Response PDU shall have Read STag in the RsvdULP field of the DDP Header
- If Write STag was advertised and Read STag was not advertised (WSV=1, RSV=0 in iSER Header) in the Send with SE Message carrying SCSI command PDU, then Send with Inv. and SE carrying SCSI Response PDU shall have Write STag in the RsvdULP field of the DDP Header
- Receive completion event corresponding to Send with Inv. and SE provided to iSER shall include the value of the invalidated STag

iSCSI-R Mapping Slide 12

IBM moves:
An iSCSI-R enabled node must initiate the iSER support negotiation during iSCSI login if it wants the connection to be in iSER mode; however it is not required to do so if it prefers the connection to be in iSCSI-v1 mode
EMC seconds.
Motion passes by acclamation.

EMC moves: **RDMAExtension key is not applicable in a discovery session and is marked "irrelevant".**
IBM seconds.
Motion passes by acclamation.

Broadcom moves: **The RDMAExtension key should be negotiated before any login parameters which may be affected by the mode selection.**
IBM seconds.
Pat: Friendly amendment should-> shall.
The RDMAExtension key shall be offered only on the initial login request PDU or login response PDU of the leading connection, and if offered the response shall be sent in the first available login response or login request PDU. The key shall precede any login parameters which may be affected by the mode selection.
Friendly amendments accepted.
Motion passes by acclamation.

HP moves: **iSER shall not provide a CRC beyond what MPA provides, and iSER requires that iSCSI must negotiate the header digest and data digest to be negotiated None.**

EMC seconds.
Motion passes by acclamation.

***Microsoft moves:* October 29, 2002 is the feature proposal freeze date for storage verbs.**
HP seconds.
IBM votes no.
Motion passes.

RSTG-mtg-021014-jc.doc

HP motions that we adopt approach A for realizing iSCSI data ACKs when the operational ErrorRecoveryLevel is 2 for an iSCSI connection.
Broadcom seconds
Motion passes by acclimation
Updated slide is as follows (Data_Ack_in_iSER_v0.3.ppt)

RDMAC SCSI Mapping

Data acks with ErrorRecoveryLevel=2

- An equivalent of iSCSI Data ACK (for connection recovery) may be provided in one of the following ways (In all approaches, targets treat either the received Send Message or the Read completion semantically identical to an explicit iSCSI data ack) –
  - A. Mandate a zero-length RDMA Read following every RDMA Write that the target wants acknowledged. A Read Response signals remote completion of all preceding RDMA Writes. The zero-length RDMA Read, if issued, must be issued before the status for the SCSI command is sent.
    - An additional operation to carry out, but requires no enhancements to RDMAP protocol.
  - B. Propose enhancements to RDMAP to automatically generate a Send Message from the Data Sink on receiving the concluding DDP Segment of an RDMA Write Message (Perhaps only for a new opcode "RDMA Write w/ response"? Or perhaps on a new bit in the RDMAP header?) that echoes the RDMAP header of that (doesn't have to be the last) DDP Segment.
    - Enhancements to RDMAP Protocol. It also does imply additional RNIC state as well.
  - C. Propose a "Message reflector" QN that simply echoes any Send message directed to it. Target sends a Send message to this QN after every RDMA Write it wants acknowledged.
    - Any protocol requiring a data ACK could employ this generic RNIC facility. But are there others?
    - Expensive to support an additional QN?
  - D. iSER on the target issues an ACK-Request Send message that carries a per-task AckSN counter. iSER on the initiator simply responds back with an ACK-Reponse echoing the same payload. By definition, all the preceding posted RDMA Writes must had completed when ACK-Request is processed, so the ACK-Response acknowledges all those.
    - Initiators running iSER in software (i.e. generic RNIC) would take an addl. Interrupt to process each ACK-Request. Also implies additional code/ state in the iSER layer to manage the data ACKs.

Votable: HP motions that we adopt approach A for realizing iSCSI data ACKs when the operational ErrorRecoveryLevel is 2 for an iSCSI connection.

V0.3 Slide 3 October 14, 2002

HP motions that the legal behavior to realize iSCSI data acks on iSER targets be as described on slides 5, 6 and 7.
Broadcom seconds
Motion passes by acclimation
Updated slides 5, 6, and 7 are as follows (Data_Ack_in_iSER_v0.3.ppt)

RDMAC SCSI Mapping

iSER realization of data acks - 1

- This slide summarizes the semantic behavior expected of iSER to realize iSCSI Data Acks, when the operational ErrorRecoveryLevel of an iSCSI connection is 1.

- On the Put_Data invocation from the iSCSI layer for a SCSI Data-In PDU with the A-bit set to 1, the iSER layer shall do the following–
  a) Originate an RDMA Write Message that moves the data contained in the data segment of the SCSI Data-In PDU to the appropriate Data Sink buffer on the initiator. The local completion event of sending this RDMA Write Message must result in a Data_Completion_Notify back to the local iSCSI layer as per the semantics defined in the DI draft.

  b) The local completion event of sending this RDMA Write Message must also result in a Data_ACK_Notify back to the local iSCSI layer as per the semantics defined in the DI draft.

V0.3 Slide 1 October 14, 2002

RDMAC SCSI Mapping

iSER realization of data acks - 2

- This slide summarizes the semantic behavior expected of iSER to realize iSCSI Data Acks, when the operational ErrorRecoveryLevel of an iSCSI connection is 2.

- On the Put_Data invocation from the iSCSI layer for a SCSI Data-In PDU with the A-bit set to 1, the iSER layer shall originate the following RDMA Operations in this order –
  a) An RDMA Write Message that moves the data contained in the data segment of the SCSI Data-In PDU to the appropriate Data Sink buffer on the initiator. The local completion event of sending this RDMA Write Message must result in a Data_Completion_Notify back to the local iSCSI layer as per the semantics defined in the DI draft.

  b) A zero-length RDMA Read Message that is directed to the same initiator STag representing the Data Sink buffer. The RDMA Read Response reception event in the iSER layer must in turn cause a Data_ACK_Notify back to the local iSCSI layer as per the semantics defined in the DI draft.

V0.3 Slide 6 October 14, 2002

iSER realization of data acks - 3

- Target iSER implementations may choose to always realize the iSCSI Data ack functionality using the model described for ErrorRecoveryLevel=2 – i.e. regardless of the operational ErrorRecoveryLevel.
- The choice to implement beyond what's minimally required for ErrorRecoveryLevel=1 could be made by a target for the following reasons: iSER code path simplicity, desire to not maintain the state of operational ErrorRecoveryLevel in iSER Note that the negotiation of the ErrorRecoveryLevel key itself is done during the iSCSI level login, **not** at the iSER level.

Votable: HP motions that the the legal behavior to realize iSCSI data acks on iSER targets be as described on slides 5, 6 and 7.

V0.3 Slide 7 October 14, 2002

HP motions to approve this slide deck (Error_handling_recover_v0.30.ppt)
Intel seconds
Motion passes by acclimation
Slide deck is as follows (Error_handling_recover_v0.30.ppt)

iSER: Error handling and Recovery

John Hufferd
Mallikarjun Chadalapaka

V0.30 Slide 1 October 10, 2002

RDMAC iSCSI Mapping

Data integrity & Error recovery

- This slide set proposes how iSCSI/ iSER should realize the error management hierarchy as defined in the iSCSI specification.
- iSER by virtue of running iWARP (unlike the original iSCSI) has the advantage that a robust (MPA) CRC-assisted error detection mechanism is available in the transport layer. In the iWARP context, it's reasonable to presume that the error rate would be extremely low (compared to original iSCSI that operated on 16-bit-checksum-assisted TCP).
- Because of this reason, the storage subteam had already unanimously voted to mandate that iSCSI/ iSER MUST negotiate HeaderDigest and DataDigest to be None.
- Given this context, the proposal is that:
  1. PDU-level recovery be **not** a design consideration for iSER. iSCSI clearly defines all the expected protocol interactions around PDU-level recovery today, it's not an iSER issue.
     - SNACK/ recovery R2T operations, (if issued inadvertently by the iSCSI layer), could be supported by iSER with no distinction (via the standard Send_Control and Get_Data Operational Primitive interface).
  2. iSCSI connection recovery must be supported.

V0.30 Slide 2 October 10, 2002

RDMAC iSCSI Mapping

ErrorRecoveryLevel semantics for iSER

- The proposed iSCSI/ iSER protocol behavior corresponds to the following error management hierarchy.

ErrorRecoveryLevel	Implies	iSER support	Comments
0	Session recovery	Yes	Default mode of operation
1	PDU recovery	Yes	Supported with no addl. cost with iSCSI/ iSER because digest/ sequence errors never happen, timeout-driven proactive SNACK SHOULD however be disabled (not useful anyway), but iSCSI already has the protocol to deal with these SNACKs if it does happen.
2	Connection recovery	Yes	Supported. iSER in conjunction with iSCSI on the target will drive all pending/ failed RDMA operations on the new connection upon reassignment (described later).

V0.30 Slide 3 October 10, 2002

RDMAC iSCSI Mapping

Connection recovery

- Connection recovery refers to reassigning the connection allegiance of an active task to a different iSCSI connection, upon the failure of the original connection the task was allegiant to.
- SCSI Write
  - At the time the original connection failed, iSCSI layer knows how many write data sequences have been successfully fetched based on which RDMA Read operations the local iSER layer successfully completed so far (as indicated by the Data_Completion_Notify invocations).
  - Upon accepting a reassignment request for a SCSI Write command, iSCSI/ iSER target issues the Get_Data Operational Primitive calls (R2T PDUs) for the pending write data sequences to be received. iSER layer translates each Get_Data into an RDMA Read operation, to eventually result in a corresponding invocation of Data_Completion_Notify.
  - Eventually, iSCSI layer normally concludes the task by sending the status.
- SCSI Read
  - An iSCSI/ iSER initiator MUST issue a task reassignment request with ExpDataSN=0 (which means "send me all unacknowledged data" to the target iSCSI layer). This is the only appropriate choice for iSCSI/ iSER initiators because the data transfer and even the data acknowledgements happen completely transparent to iSCSI initiator layer.
  - Upon accepting a reassignment request for a SCSI read command, an iSCSI/ iSER target issues Put_Data Operational Primitive calls (SCSI Data-In PDUs) for all unacknowledged data. This may result in the entire read data for the I/ O getting transferred again if data acks were **not** employed for the command. iSER layer translates each Put_Data into an RDMA Write operation, to eventually result in a corresponding invocation of Data_Completion_Notify.
  - Eventually, iSCSI layer normally concludes the task by sending the status.

V0.30 Slide 4 October 10, 2002

RDMAC iSCSI Mapping

Connection recovery (contd.)

- When a task is reassigned to a new iSCSI connection, the iSER layer on the initiator MUST specify the valid STag(s) (none, one or two) in the iSER header as applicable on the new connection as it did when the command was initially issued, and set the WSV and the RSV bits correspondingly.

- When a task is reassigned to a new iSCSI connection, the iSER layer on the target MUST consider the valid STags (none, one or two, as denoted by the WSV and RSV bits), if any, in the iSER header as the newly applicable initiator STag(s) for the task on the new connection. The target MUST discard the STag(s), if any, that have been valid for the task until that point of time.
  - In other words: Need to have iSER Target layer relate the valid STags with the ITT, replacing the previous ITT relationship always (unconditionally changes the ITT-STag association **without** the knowledge of the underlying iSCSI PDU).

V0.30 Slide 5 October 10, 2002

RSTG-mtg-021023-pat.doc

Moved by Broadcom: Queuing of outstanding RDMA Read operation beyond the peer's RDMA Read Request Queue depth should be done at iSER.
Second: HP
Passes by acclamation

RSTG-mtg-021028-jw.doc

Broadcom moves - The transition from iSCSI on TCP to iSER happens on the same connection iSCSI was negotiated on and other options are going to be eliminated.
IBM Seconds
Friendly amendment – remove "and other options are going to be eliminated"
Broadcom accepts amendment
IBM accepts amendment
Updated motion – The transition from iSCSI on TCP to iSER happens on the same connection iSCSI was negotiated on.
Motion passes by acclimation

Broadcom moves - Initiator's inbound RDMA Rd request Queue size, will be set locally and declared to the Target during iSER mode.

IBM seconds

Further Discussion

IBM friendly amendment - **Initiator's inbound RDMA Rd request Queue size, will be set locally and declared to the Target during iSER mode for each connection.**

Broadcom accepts amendment

IBM accepts amendment

Motion passes by acclimation

RSTG-mtg-021030-mk.doc

Broadcom motions: In transition from iSCSI to iSER the QP setup and resource allocation will be handled during iSCSI login negotiation
EMC seconds

Motion passes by acclimation

RSTG-mtg-021104-hvs.doc

IBM motions that we adopt the key definitions on slide 6.
EMC seconds.
Any further discussion?
No abstaintions.
Motion passes by acclamation.
Slide 6 is as follows (RDMAC_FullSendDataSegmentLength_v32.ppt):

FullSendDataSegmentLength Key (Option 2)

- Use: IO (initialize only)
- Senders: Initiator and Target
- Scope: CO (connection-only)
- Irrelevant when: RDMAExtension=No
- FullSendDataSegmentLength=<numerical-value-512-to-(2**24-1)>
- Default is 8192 bytes
- Value function is Min
- This key is relevant only for a leading connection if the RDMAExtension key is negotiated to "Yes", or for a non-leading connection where RDMAExtension was previously negotiated to "yes". It is used by the initiator and the target to negotiate the size of the data segments in all non-last iSCSI PDUs containing unsolicited data. The initiator MUST send all non-last iSCSI PDUs containing unsolicited data with a data segment of exactly FullSendDataSegmentLength size whenever the PDUs constitute a data sequence whose size is larger than FullSendDataSegmentLength.

Also, add to MaxRecvDataSegmentLength:

- Irrelevant when: RDMAExtension=Yes

iSCSI-R Mapping Slide 6

RSTG-mtg-021106-pat.doc

HP moves that we accept the iSER-floating-credit-v0.5.ppt as the flow control mechanism for iSER.
Broadcom seconds.
Passes by acclamation
Slide deck is as follows (iSER-floating-credit-v05.ppt):

iSER Send Type Message Floating-Credit Flow Control v0.5

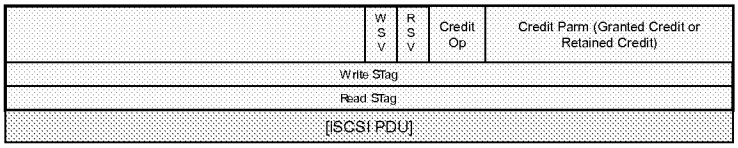
Jim Wendt
Hemal Shah
Mallikarjun Chadalapaka
Pat Thaler
November 6, 2002

Floating-Credit Flow Control Summary

1) iSER shall flow control all Send Type Messages (Untagged Buffers) between Data Source and Data Sink
2) iSER shall flow control in both directions (Initiator-to-Target and Target-to-Initiator)
3) iSER flow control shall be per iSER Connection (e.g. RDMAP Stream)
4) One flow control credit shall correspond to one Send Type Message (e.g. one Untagged Buffer)
5) One Send Type Message sent from Data Source to Data Sink shall consume one credit
6) Data Sink shall grant credits to the Data Source by sending an additive number of credits to the Data Source.
7) Data Source shall maintain an absolute count of available credits at the Data Sink for receiving Send Messages
8) Data Source shall reserve its last available credit for granting credit to the Data Sink (I.e. reverse direction credits)
9) Data Sink can request the Data Source to return its unused credits except for a specific number of retained credits which must be at least one. If the Data Source has less than the retained number of credits then no credits are returned.
10) Data Sink may return unused credits to the Data Source by sending one zero-length Send Type Message for each credit being returned to Data Sink
11) The last Send Type Message in a series of returned unused credits should be a SendSE or SendInvSE
12) During initialization, a Data Sink shall post at least one untagged receive buffer large enough to receive an initial message from the Data Source granting credit

Slide 2

Flow Control Fields in iSER Header



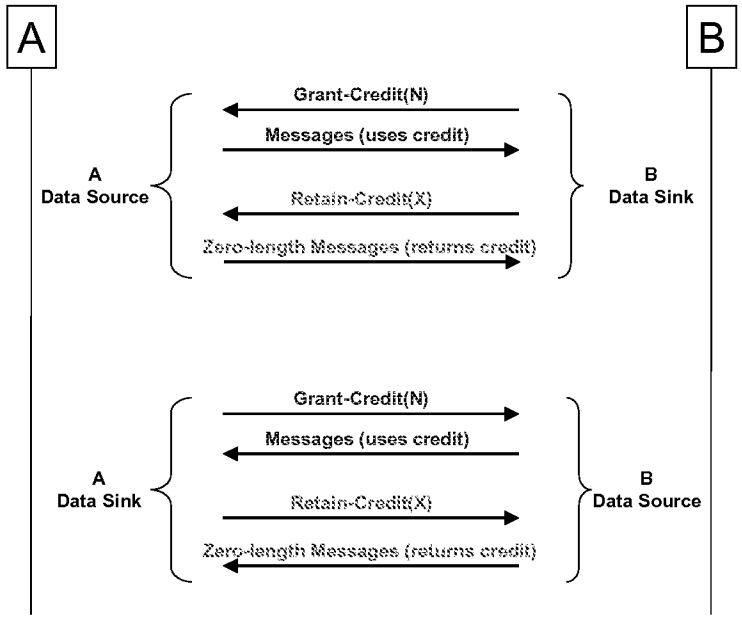
Note: The final location of the credit mgmt fields in the iSER header are TBD

Credit Op subfield (2 bits)	Credit Parm subfield (12 bits)
00 = No credit operation	Sender sets to 0 / receiver ignores
01 = Grant-Credit	Granted Credit = Number of additional (additive) credits given to the Data Sink by the Data Source (1 to 2^12-1)
10 = Retain-Credit	Retained Credit = Maximum number of absolute credits to be retained by the Data Source after returning all other credits to the Data Sink (1 to 2^12-1)
11 = reserved	reserved

- If no iSCSI PDU (or iSER control PDU) is being sent in this message, then
  - The length of the iSER PDU is the length of the iSER fixed header
  - The Write Stag and Read Stag fields are N/A (WSV=0 / RSV=0)

Slide 3

Flow Control Exchanges



Slide 4

Grant Credit - Data Sink to Data Source

- Grant-Credit (Data Sink to Data Source)
  - Used by the Data Sink to grant credit to the Data Source
  - The number of credits granted to the Data Source is an additive amount
  - Data-Sink actions:
    - Set iSER Header <Credit Op> to "Grant-Credit"
    - Set iSER Header <Granted Credit> to number of additional credits given to Data-Source
    - If piggybacking on an existing PDU going from Data Sink to Data Source:
      - Transfer iSER PDU to Data Source
    - If sending as a separate message from Data Sink to Data Source:
      - Set iSER Header WSV and RSV to zero
      - Transfer iSER Header-only PDU to Data Source
  - Data-Source actions:
    - Add <Granted Credit> value to an absolute count of available credits (Max-Retained-Credits) on Data-Sink
    - The maximum absolute count of credits which the Data Source can accumulated is limited to 2^12-1

Slide 5

Use Credit - Data Source to Data Sink

- Use Credit (Data Source to Data Sink)
  - One credit is used for each Send Type Message transferred from the Data Source to the Data Sink
  - Data Source actions:
    - Decrement by one the absolute count of available credits for sending messages to the Data Sink
    - Data Source shall only use its last available credit to the Data Sink if it is granting more than zero credit to the Data Sink (I.e. reverse direction credits)
  - Data Sink actions:
    - Receive one used credit from the Data Source

Slide 6

Retain Credit – Data Source to Data Sink

- Retain-Credit Request (Data Sink to Data Source)
  - Used by the Data Sink to request that the Data Source retain a certain number of credits and return the remaining credits currently owned by the Data Source to the Data Sink
  - Data Sink actions:
    - Set iSER Header <Credit Op> to "Retain Credit"
    - Set iSER Header <Retained Credits> to number of credits the Data Source is to retain after returning all other credits to the Data Sink
    - If piggybacking on an existing PDU going from Data Sink to Data Source:
      - Transfer iSER PDU to Data Source
    - If sending as a separate message from Data Sink to Data Source:
      - Set iSER Header WSV and RSV to zero
      - Transfer iSER Header-only PDU to Data Source
    - Data Sink must allow at least one retained credit on the Data Source (for sending a reverse channel Grant Credit message)
  - Data Source actions:
    - Send all currently owned credits (at time of processing the Retain-Credit) less the number of <Retained Credits> to the Data Sink by sending one zero-length Send Message or one other Send Type Messages containing an iSER PDU for each credit being returned to the Data Sink
    - If currently owned credits are less than <Retained Credits> then no credits are returned
    - Reduce the absolute count of available credits (Max-Retained-Credits) on Data-Sink by number of credits returned

Slide 7

Return Credit – Data Source to Data Sink

- Return Credit (Data Source to Data Sink)
  - The Data Source may returns unused credits to the Data Sink by sending zero-length Send Messages
  - Data Source actions:
    - Send one zero-length Send Message or one Send Type Message with an iSER PDU to the Data Sink for each credit being returned to the Data Sink
    - The last Send Type Message in a series of returned unused credits should be a SendSE or SendInvSE
    - The Data Source should return unused credits upon receiving the Retain-Credit request rather than waiting for the need to send an actual iSER PDU
  - Data Sink action:
    - Receive one returned credit from the Data Source for each Send Type Message

Slide 8

Bidirectional Flow Control

- Each node functions as both a Data Source and a Data Sink
- All flow control operations occur in both directions between two nodes
- All messages, including flow-control messages, consume one credit at the receiving node, including:
  - iSER Header-only Grant-Credit messages
  - iSER Header-only Retain-Credit messages

Slide 9

Initialization

- At initialization, the Data Sink shall post one untagged receive buffer for receiving an initial Grant Credit from the Data Source

Slide 10

RSTG-mtg-021115-jw.doc

HP motions that we accept the stg_iSER-connect-v07.ppt slide deck with the following change – Replace all occurrences of MAY/MUST with MAY, there are three such occurrences in the slide deck, and accept option.2 on slide 3.

Broadcom seconds

Any objections, discussion, abstentions? None.

Motion passes with acclamation.

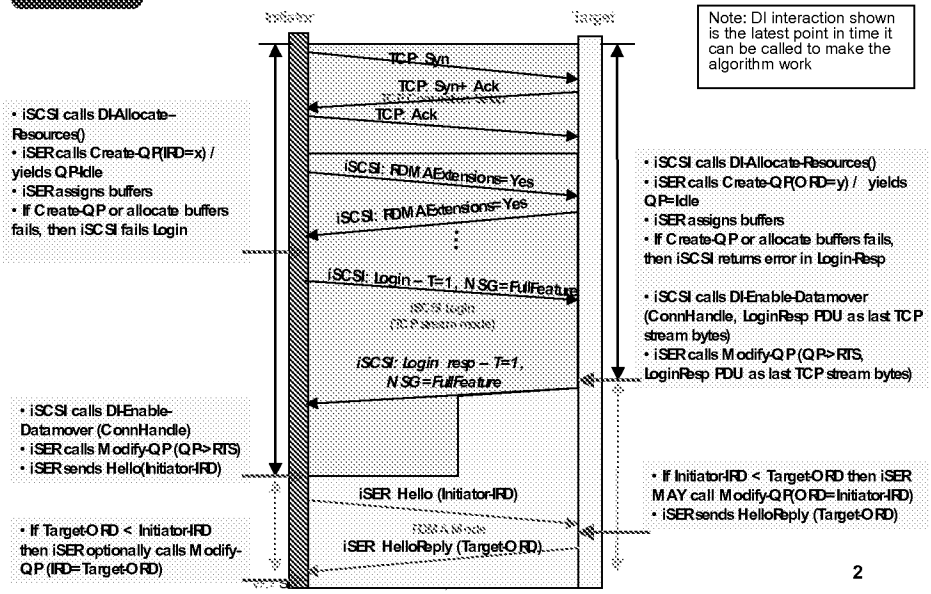
<Updated slide deck: stg_iSER-connect-v1.ppt>

iSER Connection Setup
v1

Uri Elzur - Broadcom
Jim Wendt - HP
Mallikarjun Chadalapaka - HP

RDMAC iSCSI Mapping

iSER Connection Setup



2

RDMAC iSCSI Mapping

Options for Target's setting of ORD

- If Initiator-IRD < Target-ORD then:

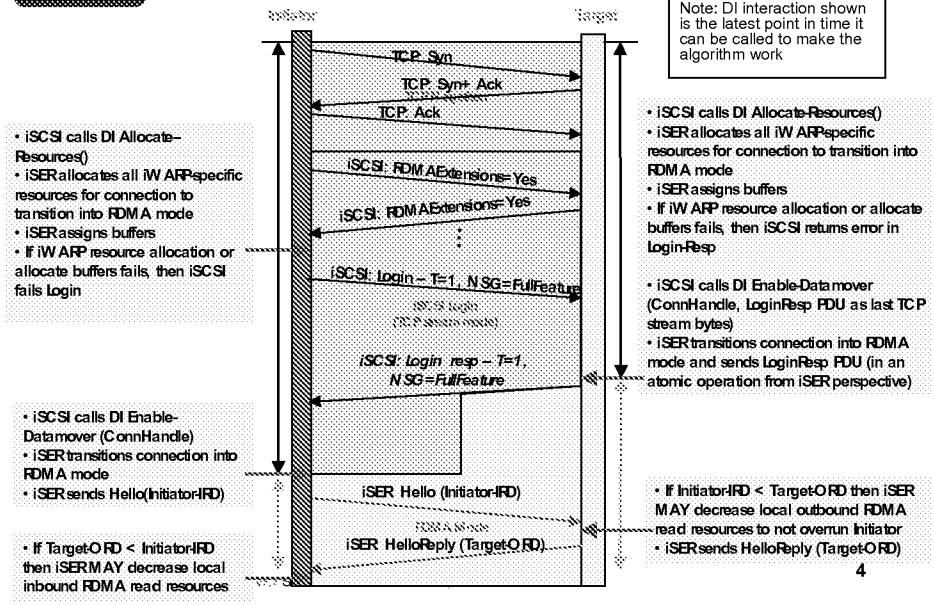
1) iSER **MUST** call Modify-QP(ORD=Initiator-IRD)
   - Pro:
     - Provides an additional safeguard (at the RNIC level) that Initiator-IRD won't be exceeded
   - Con:
     - **Requires that all target implementations have the ability to modify ORD on RNIC (specifically when QP is in RTS state)**
     - It is unnecessary to modify the target's ORD because iSER ensures that no more than Initiator-IRD RDMA Read Requests will be issued to the RNIC at any given time

2) iSER **MAY** call Modify-QP(ORD=Initiator-IRD)
   - Pro:
     - Implementations don't have to support modification of ORD on RNIC
     - Specifies only the minimum required behavior for the protocol
   - Con:
     - Less protection against a faulty iSER implementation (minor)

Voted on 11/15/02 to accept option 2)

RDMAC iSCSI Mapping

iSER Connection Setup – Verb-Free Version



4

RDMAC iSCSI Mapping

Initiator-IRD and Target-ORD Handling

- Initiator
  - Send Initiator-IRD in Hello to Target
- Target
  - If Initiator-IRD < Target-ORD
    - then Target iSER MAY call Modify-QP(ORD=Initiator-IRD)
  - Send Target-ORD in HelloReply to Initiator
- Initiator
  - If Target-ORD < Initiator-IRD
    - then Initiator iSER MAY call Modify-QP(IRD=Target-ORD)

- Target
  - Target MUST NOT issue more RDMA Read Requests than the value of Initiator-IRD as agreed to by both sides after the Hello and HelloReply exchange

5

IBM motions that we accept the slide deck (Init_time_Datamover_Interactions_v0.30.ppt)

HP seconds
Further discussion - none
Vote against - none

Abstain – none
Motion passes by acclimation

<Slide deck: Init_time_Datamover_Interactions_v0.30.ppt>

Init-time & Shutdown-time - Datamover Interfaces

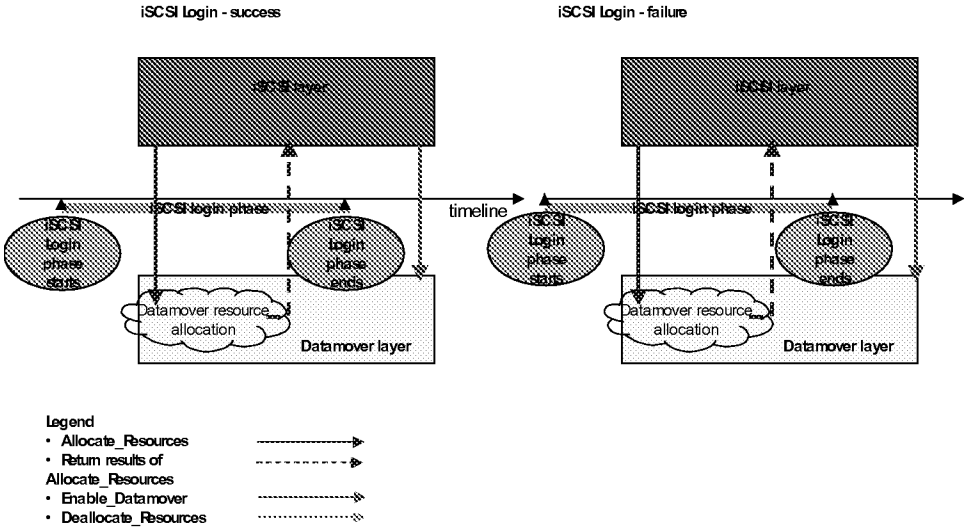
John Hufferd
Mallikarjun Chadalapaka

Problem statement

- The storage subgroup had already taken the binding vote that the conclusion of iSCSI login phase (i.e. the final Login Response in streaming mode from the iSCSI target) transitions the connection into the RDMA mode.
- At least one of the resources that ought to be acquired by both sides before signaling the Login as complete is iWARP-specific, the QP (with an associated RRQ depth). This is a resource that is completely in the knowledge domain of the Datamover layer, but ideally should be allocated before the iSCSI login phase completes.
- The current DI draft (rev 0.5) does not define Operational Primitives to cater to these init time (aka "pre-iSER") interactions between the iSCSI layer and Datamover layer.
- We can choose to not model these interactions in DI, to leave them as implementation-specific. The downside is that DI will not then reflect the minimally required architectural requirements for Datamover operation. Besides, this resource allocation is really a generic issue that any hypothetical Datamover (other than iSER) would need to address. But, we should be careful **not** to over-specify all implementation-specific exchanges that happen between the layers (such as error handling etc.) that aren't pertinent to Datamover protocol operation.
- **Votable: HP motions that the init time interactions between iSCSI and a Datamover be modeled in DI.**

RDMAC iSCSI Mapping

Initiator's usage: Proposed new Primitives



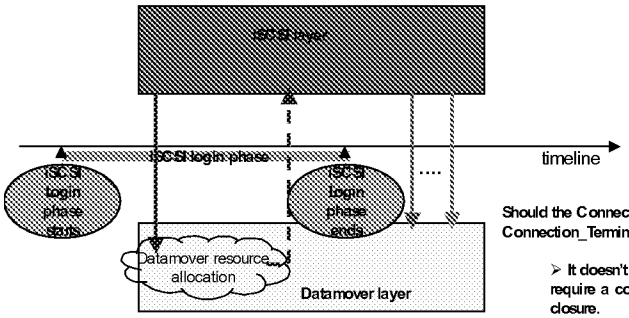
V0.30 Slide 3 November 04, 2002

RDMAC iSCSI Mapping

Initiator's usage: Proposed new Primitives

iSCSI full-feature phase - termination

In addition to the problem statement earlier stated, it also turns out that we need to model the iSCSI connection shutdown-time interactions in DI.



Should the Connection_Terminate Operational Primitive result in a Connection_Terminate_Notify (that DI already defines)?

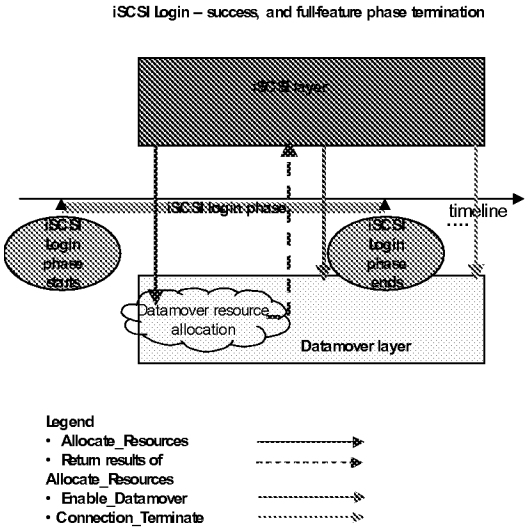
- It doesn't seem necessary, iSCSI state diagrams don't require a confirmation for a locally-initiated TCP connection closure.
- Connection_Terminate_Notify is still, however, required to notify the iSCSI layer of the connection termination events caused by remote actions.

Legend
• Allocate_Resources
• Return results of Allocate_Resources
• Enable_Datamover
• Connection_Terminate

V0.30 Slide 4 November 04, 2002

RDMAC iSCSI Mapping

Target's usage: Proposed new Primitives



RDMAC iSCSI Mapping

Proposed semantics

- Allocate_Resources:
  - Input qualifiers are: Connection_Handle, Resource_Descriptor (implementation-dependent, i.e. structurally opaque to DI)
  - This Operational Primitive is meant to request the Datamover layer that it perform all the Datamover-specific resource allocations required for an operational iSCSI connection. The handle identifies the connection the iSCSI layer is requesting the resource allocation for, to eventually transition the connection to be a Datamover-accelerated iSCSI connection.
  - Return results: Status
  - Status=0 means that the Allocate_Resources invocation corresponding to that Connection_Handle succeeded. Status=1 means that the Allocate_Resources invocation corresponding to that Connection_Handle failed. There MUST NOT be more than one Allocate_Resources Primitive outstanding for a given Connection_Handle at any time.

- Enable_Datamover:
  - Input qualifiers are:
    Connection_Handle, TCP_Connection_Descriptor [, Final_Login_Response_PDU]
  - This Operational Primitive requests the Datamover layer to accelerate all further iSCSI exchanges on the iSCSI connection identified by the Connection_Handle, for which the Datamover resource allocation was earlier made. The TCP connection associated to the Connection_Handle is specified in the TCP_Connection_Descriptor (aka TCB).
  - The Final_Login_Response_PDU input qualifier is valid only for a target, and contains the final Login Response which must be sent as a TCP byte stream, as expected by the traditional iSCSI, before the Datamover acceleration is enabled for the TCP connection.

Proposed semantics

- TCP_Connection_Descriptor is an information element that identifies a specific TCP connection to the Datamover layer. The exact structure of this Descriptor is implementation-dependent.

- Deallocate_Resources:
  - Input qualifier is: Connection_Handle
  - This Operational Primitive is meant to request the Datamover layer that it deallocate all the Datamover-specific resource allocations that were earlier made for the Connection_Handle.

- Connection_Terminate:
  - Input qualifier is: Connection_Handle
  - This Operational Primitive requests the Datamover layer to terminate the LLP connection and deallocate all the resources associated with the Connection_Handle.

RSTG-mtg-021204-TT.doc

IBM moves we reopen the discussion of the FullSendDataSegmentLength in the 11/04 vote.

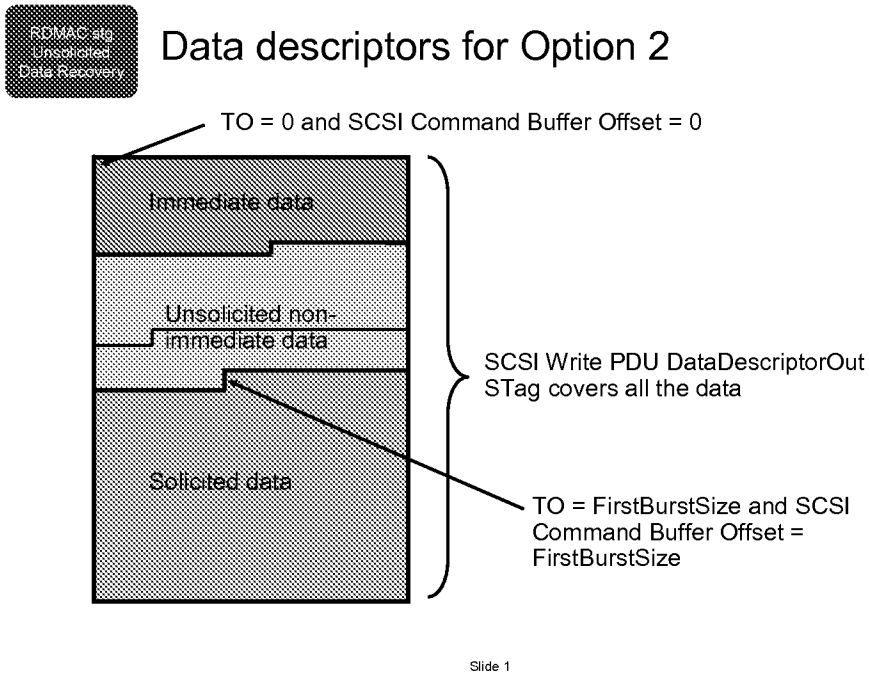
Intel seconds
Vote against - none
Abstain - none
Absent - Agilent
Motion passes by acclimation

IBM moves that we overturn the vote on 11/4 entitled "FullSendDataSegmentLength Key (Option 2)".

Intel seconds
Vote against - none
Abstain - none
Absent - Adaptec
Motion passes

Agilent moves that Option 2 be adopted : "SCSI Writes use one data descriptor".

IBM seconds
Vote against - none
Abstain - Adaptec
Motion passes



RSTG-mtg-021209-jc.doc

IBM Motions to accept option 3 and the related keys as presented in slide 7, 8, 9, & 10 of the slide deck (senddatasegmentlength_v3.ppt)

Agilent seconds
Vote against - none
Abstain - Adaptec, NetApp
Motion passes

Option 3: The New Key Replaces MRDSL Only for iSCSI Command PDUs & iSER Data-type PDUs

- Only SCSI Command PDUs and SCSI Data-out PDUs are affected at the initiator
  - Existing iSCSI implementation for other iSER control-type PDU besides SCSI Command PDU and SCSI Data-out PDU continue to use MRDSL at the initiator side
    - The iSCSI layer need not send all control-type PDUs with a size exactly equal to FSDSL if the size of the data sequence exceeds FSDSL
  - The other control-type PDUs are not used as frequently as SCSI Data-out
    - Their impact on receive buffer memory management is not as critical
- Only SCSI Data-in PDUs are affected at the target
  - Existing iSCSI implementation for other iSER control-type PDU besides SCSI Data-in PDU continue to use MRDSL at the initiator side
    - The iSCSI layer need not send all control-type PDUs with a size exactly equal to FSDSL if the size of the data sequence exceeds FSDSL
  - The other control-type PDUs are not used as frequently as SCSI Data-in
    - Their impact on receive buffer memory management is not as critical
- Proposal:
  - Change FSDSL to TargetRecvDataSegmentLength for the initiator
  - Add InitiatorRecvDataSegmentLength for the target
  - TargetRecvDataSegmentLength supercedes target MRDSL in the full feature phase for SCSI Data-out PDUs only
  - InitiatorRecvDataSegmentLength supercedes initiator MRDSL in the full feature phase for SCSI Data-in PDUs only
  - Target MRDSL at the target and initiator would be set to the value of TargetRecvDataSegmentLength in the full feature phase
  - Initiator MRDSL at the initiator and target would be set to the value of InitiatorRecvDataSegmentLength in the full feature phase
- Pro:
  - Less impact to existing iSCSI implementation
- Con:
  - Potentially less than optimal receive buffer memory utilization for the other iSER control-type PDUs which are affected by MRDSL

Slide 1

TargetRecvDataSegmentLength Key

- Use: IO (initialize only)
- Senders: Initiator and Target
- Scope: CO (connection-only)
- Irrelevant when: RDMAExtensions=No
- TargetRecvDataSegmentLength=<numerical-value-512-to-(2**24-1)>
- Default is 8192 bytes
- Result function is Min
- This key is relevant only for the iSCSI connection of an iSCSI session if RDMAExtensions=Yes was negotiated in the leading connection of the session. It is used by the initiator and the target to negotiate the size of the data segments in SCSI Command PDUs and SCSI Data-out PDUs for non-Final iSCSI PDUs to be sent by the initiator. The initiator MUST send all such non-last iSCSI PDUs with a data segment of exactly TargetRecvDataSegmentLength size whenever the PDUs constitute a data sequence whose size is larger than TargetRecvDataSegmentLength.

Slide 1

InitiatorRecvDataSegmentLength Key

- Use: IO (initialize only)
- Senders: Initiator and Target
- Scope: CO (connection-only)
- Irrelevant when: RDMAExtensions=No
- TargetDataSegmentLength=<numerical-value-512-to-(2**24-1)>
- Default is 8192 bytes
- Result function is Min
- This key is relevant only for the iSCSI connection of an iSCSI session if RDMAExtensions=Yes was negotiated in the leading connection of the session. It is used by the initiator and the target to negotiate the size of the data segments in SCSI Data-in PDUs for non-Final iSCSI PDUs to be sent by the target. The target MUST send all such non-last iSCSI PDUs with a data segment of exactly InitiatorRecvDataSegmentLength size whenever the PDUs constitute a data sequence whose size is larger than InitiatorRecvDataSegmentLength.

Slide 1

Changes to MaxRecvDataSegmentLength Key

- The following will be added to the key:
  - In the full feature phase for the iSCSI connection of an iSCSI session if RDMAExtensions=Yes was negotiated in the leading connection of the session, the following applies. Firstly, the key is irrelevant for SCSI Command PDUs, SCSI Data-out PDUs, and SCSI Data-in PDUs. Where MaxRecvDataSegmentLength is used as applicable to an initiator in [iSCSI] for the SCSI Command PDU and the SCSI Data-out PDU, TargetRecvDataSegmentLength MUST be used instead. Where MaxRecvDataSegmentLength is used as applicable to a target for the SCSI Data-in PDU, InitiatorRecvDataSegmentLength MUST be used instead. Secondly, MaxRecvDataSegmentLength need not be declared in the login phase. Instead in the full feature phase, the value of the initiator MaxRecvDataSegmentLength is set to InitiatorRecvDataSegmentLength in both the initiator and the target, and the value of the target MaxRecvDataSegmentLength is set to TargetRecvDataSegmentLength in both the initiator and the target.

Slide 1

RSTG-mtg-021216-ue.doc

Co-Chair selection

```
Pat - chaired 10Base and IEEE for 3 year. Lots of IETF experience.
Major challenge make IETF feel iSER is a small change and not a new
protocol.
```

Bill Edwards - no chair experience. Has experience chairing for non-profit organization. Working on iSCSI 1.0 for 2 years @HP. This is the first committee experience outside of HP.

VOTE:
Called by Uri

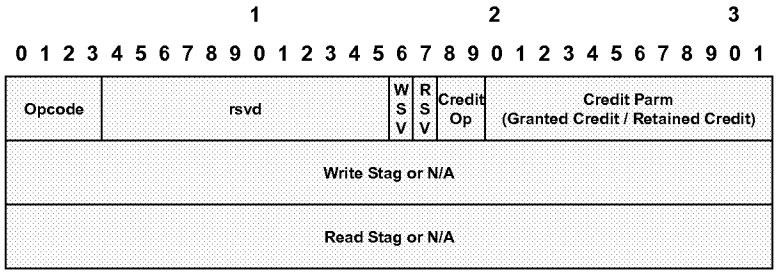
Adaptec	Pat
Agilent	Pat
Broadcom:	Pat
EMC	Pat
HP	Pat
IBM	Pat
Intel	Pat
Network Appliance	Pat

Unanimous vote.

HP motions that the iSER Header and Message formats be as defined in Option D on slides 17 through 21 of this slide deck (iSER-header-v06.ppt)

IBM seconds
Vote against - none
Abstain - none
Motion passes

Option D - iSER Header Format

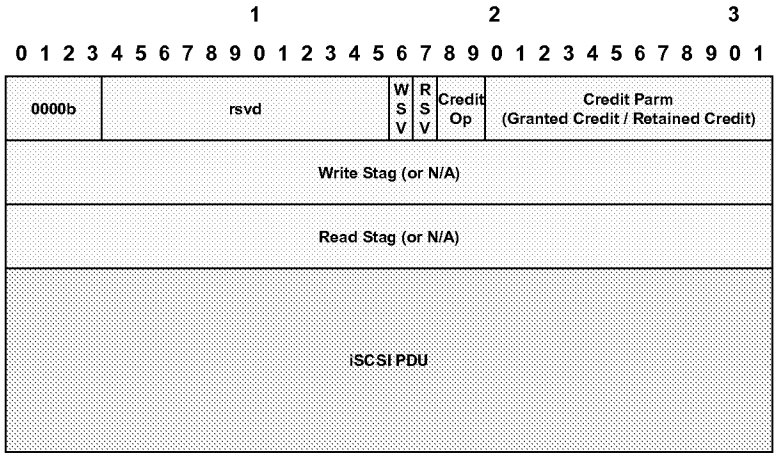


- **Opcode values:**
  - **0000b = iSCSI-PDU**
  - **0001b = iSER-NOP**
  - **0010b = Hello**
  - **0011b = HelloReply**

- **Credit Op and Credit Parm Fields already voted as part of iSER-floating-credit-v05.ppt**

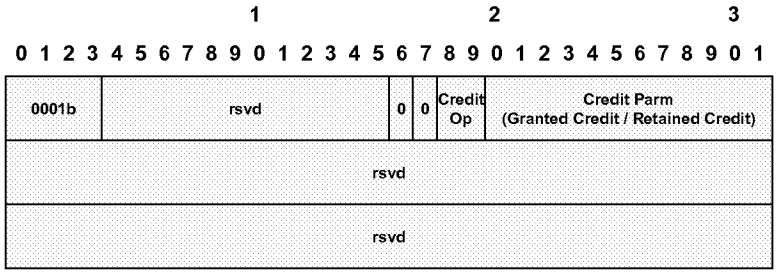
Slide 1

Option D – Opcode=0000b (iSCSI-PDU)



Slide 1

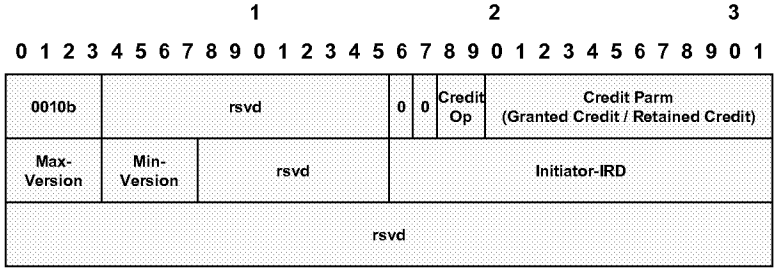
Option D – Opcode=0001b (iSER-NOP)



If no iSER message traffic is already available for sending a Grant-Credit or Retain-Credit operation, then an iSER-NOP Message is sent with the Credit-Op and Credit-Parm fields set accordingly.

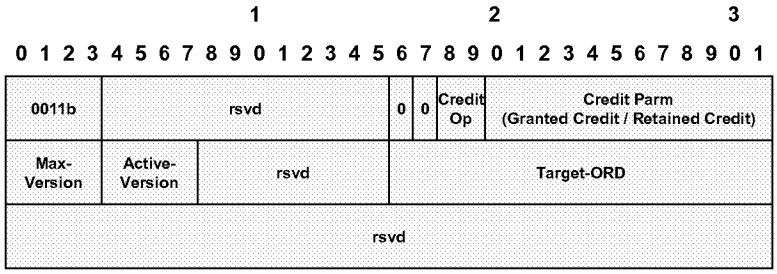
Slide 1

Option D – Opcode=0010b (Hello)



Slide 1

Option D – Opcode=0011b (HelloReply)



Slide 1

RSTG-mtg-030106-bill.doc

HP motions: Verbs should support a shared RQ (described as P-RQ in the preceding slides) across multiple QPs.

IBM seconds
Vote against - none
Abstain - none

Absent - NetApp
Motion passes

RSTG-mtg-030115.doc

HP moves to overturn the motion voted on 11/06/02 in which "HP moves that we accept the iSER-floating credit-v0.5.ppt as the flow control mechanism for iSER". HP makes this motion with the intent that the new proposal will take into account John Carrier's suggestion to add a new DI interface to let iSCSI know of the low water mark.

IBM seconds

Adaptec	Y
Agilent	Y
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Network Appliance	Abstain

Motion passes unanimously

HP moves to adopt option 4 in the iSER_flow_control_discussion_v0.22.ppt slide deck to "Completely drop the iSER Send Message Flow Control".

IBM seconds
Vote Against - NetApp
Abstain - Adaptec, EMC

Motion passes 5-1

RSTG-mtg-030122.doc

Broadcom moves that Verbs is required to support a low Watermark in the SRQ and the RI /RNIC is required to quantify the number of available WQEs on the SRQ, compare it to the low Watermark and generate an unaffiliated asynchronous event when SRQ occupancy drops below the Watermark.

EMC seconds
Vote against - none
Abstain - none
Motion passes

RSTG-mtg-030127-ue.doc

IBM motions that

A. Storage group guides the verbs writers to not add any text for setting the RQ depth when the QP is associated with a SRQ.

B. There will be a "Soft LIMIT" (smaller or equal to the SRQ depth) per QP to set a cap on the number of outstanding incomplete messages received per QP.

C. When the "Soft LIMIT" is reached the ULP is notified by an affiliated asynchronous event (one shot) and the QP continues normal receive operation.

Broadcom seconds
Vote against - Intel
Abstain - Adaptec, NetApp
Absent - EMC
Motion passes

IBM motions that we overturn the vote taken on December-16-2002 in which "HP motions that the iSER header and message format be as defined in option D on slide 17 through 21 of this slide deck (iSER-header-v0.6.ppt)"

Broadcom seconds

Adaptec	Yes
Agilent	Yes
Broadcom:	Yes
EMC	Absent
HP	Yes
IBM	Yes
Intel	Yes
Network Appliance	Yes

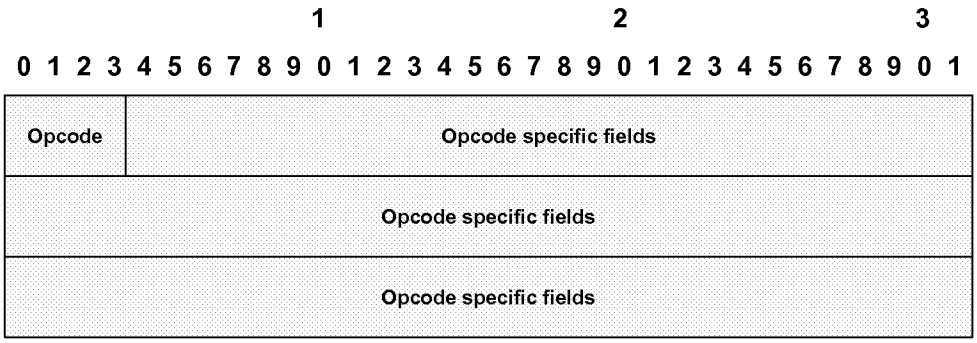
Motion passes

RSTG-mtg-030129-tc.doc

IBM motions that the iser header format be defined in option g

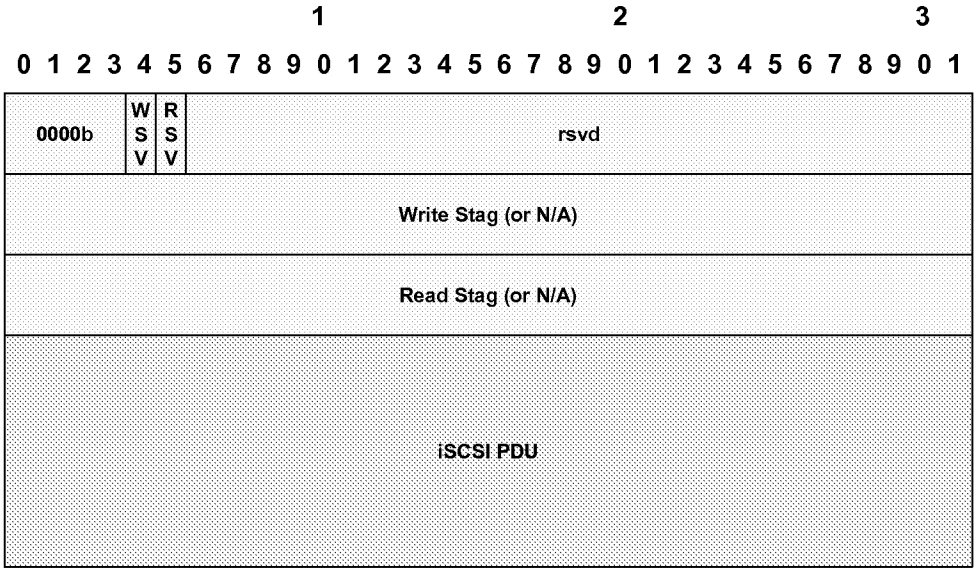
Intel seconds
Vote against - None
Abstain - None
Motion passes

Option G - iSER Header Format

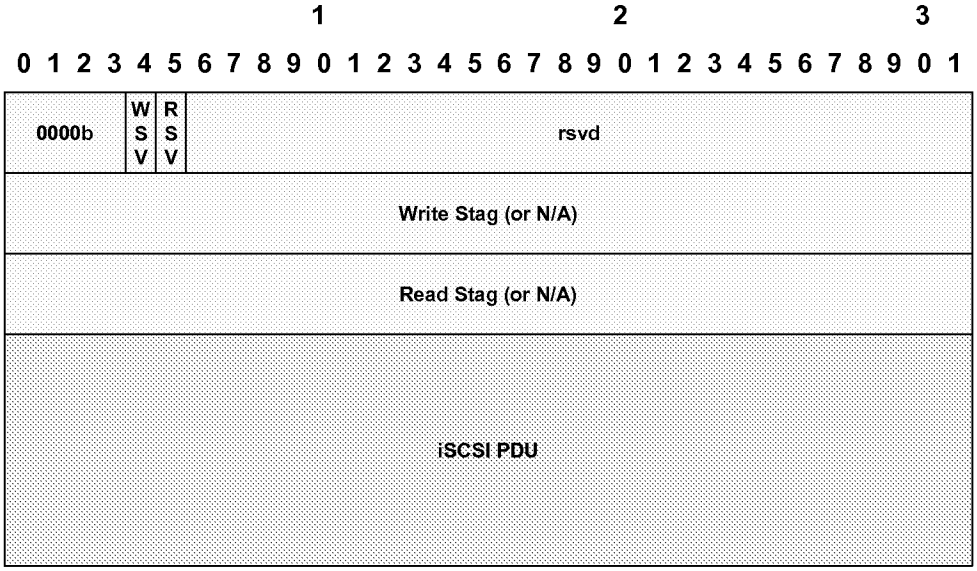


- **Opcode values:**
  - **0000b = iSCSI-PDU**
  - **0001b = Hello**
  - **0010b = HelloReply**

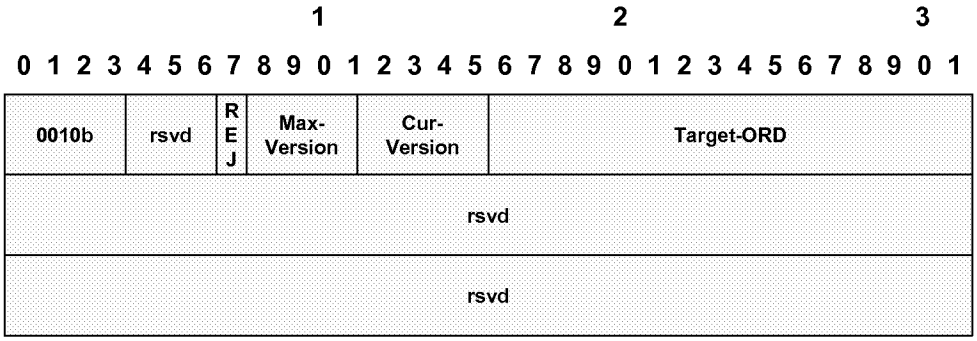
Option G – Opcode=0000b (iSCSI-PDU)



Option G – Opcode=0000b (iSCSI-PDU)



Option G – Opcode=0010b (HelloReply)



- **REJ – Reject connection, if set to '1'**

HP motions that the semantics of the new iSER header be the following. When the target determines its supported version is out bounds with respect to max version and min version specified in the hello message it MUST set the reject bit in the hello reply to one to indicate a failed hello exchange, and set the cur_version field to the lowest version it can support. The target must always set the max version field to the highest version the target can support. The Target MUST then follow up this

reponse by gracefully closing the TCP connection. The initiator iSER layer must not send any iSCSI full feature PDU's (either control or data types) until the hello reply message is received with a reject bit set to zero.

IBM seconds
Vote against - None
Abstain - None
Motion passes

RSTG-mtg-030203-Bill.doc

IBM motions to reopen a discussion on piggybacked status a.k.a. Status Phase collapse issue.
EMC seconds

Adaptec	Y
Agilent	Abstain
Broadcom:	Abstain
EMC	Y
HP	Abstain
IBM	Y
Intel	Abstain
Network Appliance	Absent

Motion passes

IBM Motions to overturn the previous status phase collapse vote to be replaced with: the ISCSI layer can use status phase collapse and if done the iSER layer must support it, by sending the data with an RDMA write followed by a Send of a zero length Data segment in a DATA IN PDU with S-bit Set.
None seconds
Motion rejected

RSTG-mtg-030210-hemal.doc

HP motions that the initiator and target STag invalidation semantics and the new DI primitive semantics proposed in this slide deck be accepted.
IBM seconds
Objections - None
Abstentions - NetApp
Motion passes

Normal task completion

- It is desirable to require the initiator implementations to cross-check the STag invalidation performed via the SendInvSE Message.
  - It does not place any new additional Verbs requirements – Verbs 0.87 (Poll CQ) text already specifies the "STag which was Invalidated" to be part of Work Completion.
  - It addresses the security concerns that are likely to be brought up on an STag invalidation requirement, where the STag value is specified by the remote node.
  - There is also the risk that the buffer pointed to by the STag is already used by iSCSI and SCSI layers (or the buffer could be in use by the ULPs).
- The proposed semantics are:

"When a unidirectional command completes on receiving a SCSI Response PDU on the initiator, the valid STag for the command, if any, is automatically invalidated. iSER layer on the initiator in such a case MUST, via RI-defined means, ensure that the correct STag is invalidated by the SendInvSE Message. If the iSER layer on the initiator deems that a wrong STag was invalidated by the SendInvSE Message, it MUST invalidate the correct STag and terminate the iSCSI connection."

"When a bidirectional command completes on receiving a SCSI Response PDU on the initiator, the valid Read STag is automatically invalidated. iSER layer on the initiator MUST manually invalidate the Write STag for the command, if any. In addition, iSER layer on the initiator MUST, via RI-defined means, ensure that the correct Read STag is invalidated by the SendInvSE Message. If the iSER layer on the initiator deems that a wrong STag was invalidated by the SendInvSE Message, it MUST invalidate the correct STag and terminate the iSCSI connection."

Abnormal termination

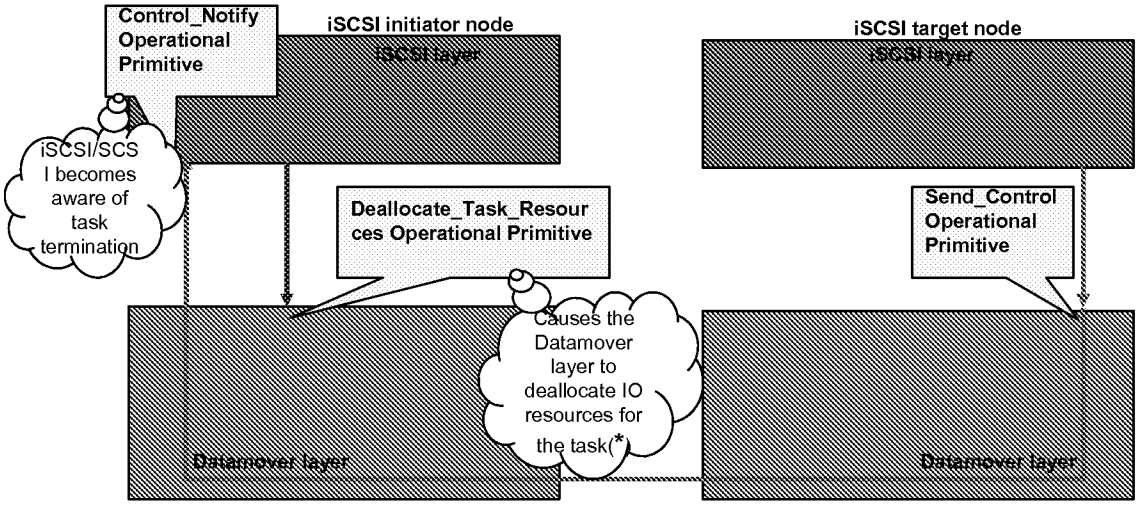
- The following cases could lead to SCSI and iSCSI layers concluding that a task terminated abnormally.
  - Completion of one of the task management functions (TMF) affecting the task in question –
    - ABORT TASK, ABORT TASK SET, CLEAR TASK SET, LU RESET, TARGET WARM RESET, TARGET COLD RESET
  - Any other iSCSI/SCSI-discernible condition (such as POWER ON RESET sense data)

- Two options to specify STag invalidation requirements are –
  a) Expect the iSER layer to detect and invalidate the STag(s) for a terminated task when appropriate.
     - Requires the iSER layer to be cognizant of the iSCSI session affiliation, and requires a lot more iSCSI/SCSI knowledge on the part of the iSER layer.

  b) Define a new DI Operational Primitive from the iSCSI layer to the iSER layer that is invoked when the task resources are to be deallocated.
     - Requires a new DI Primitive to be added, but is architecturally clean.

Proposal is to adopt Option (b).

Deallocate_Task_Resources



(*) In the case of iSER, the iSER layer invalidates the STag(s) associated with the task.

Input qualifiers to the Deallocate_Task_Resources: Connection_Handle, ITT

Return Results: Not specified

If we adopt this, it would be desirable to rename the existing Deallocate_Resources Primitive to Deallocate_Connection_Resources and Allocate_Resources to Allocate_Connection_Resources, to avoid confusion.

Abnormal termination semantics for iSER

- If we agree to define the new DI Primitive Deallocate_Task_Resources, the iSER semantics should be –

"When a unidirectional or bidirectional command terminates from the iSCSI perspective without an associated SCSI Response PDU received at the initiator, as indicated through the Deallocate_Task_Resources Primitive invocation, initiator iSER layer MUST manually invalidate the STag(s) currently associated with the task identified in the Deallocate_Task_Resources Primitive invocation from the iSCSI layer."

Target STag invalidation semantics (Sink)

- A target iSER layer implicitly Advertises its Data Sink STag when it does an RDMA Read to fetch the SCSI Write data. iSER spec rev0.75 does not define the invalidation requirements on the target after the RDMA Read Response is received (i.e. when Data_Completion_Notify is notified to the iSCSI layer).

- There are at least two options to address this.
  a) Require that the iSER layer MUST invalidate the Data Sink STag(s) at the conclusion of the I/O.
     - If an implementation strictly does this, this exposes the STag longer than it is meant for. There is also the risk that the buffer pointed to by STag is already used by iSCSI and SCSI layers (or the buffer could be in use by the ULPs).
  b) Require that the iSER layer MUST invalidate each Data Sink STag at the conclusion of the RDMA Read Operation referencing the Data Sink STag.

Proposal is to adopt (b).

Note: The existing "`RDMA Read with Invalidate Local STag`" Operation type in the PostSQ WR of RDMAC Verbs 0.87 does (b), so it does not add any additional complexity to the iSER layer to invalidate the Sink STag.

Target STag invalidation semantics (remote)

- The remote STag invalidation semantics in the normal completion are proposed to be (I think we all agree to this)

"The remote STag (i.e. initiator STag/STags to ITT) mapping MUST be maintained through the duration of the task by the target iSER layer, and the mapping MUST be invalidated by the target iSER layer upon sending the associated SCSI Response PDU for the task."

- It turns out that the Deallocate_Task_Resources Primitive proposal is also useful for a target. If we adopt the Deallocate_Task_Resources Primitive proposal, the abnormal termination semantics can be summarized thus -

"When a unidirectional or bidirectional command terminates from the iSCSI perspective without an associated SCSI Response PDU sent from the target, as indicated through the Deallocate_Task_Resources Primitive invocation, target iSER layer MUST invalidate the remote STag(s) currently associated with the task identified in Deallocate_Task_Resources Primitive invocation from the iSCSI layer."

RSTG-mtg-030212jc.doc

IBM motions that we adopt the schedule as shown in this doc (stg_doc_schedule_v03.ppt)
HP seconds
Objections - None
Abstentions - None
Motion passes

iSER/DI Specification Schedule

	Schedule worked out at 9/4 F2F	Revised schedule for iSER on 2/11/03	Revised schedule for DI on 2/11/03
.1 Annotated Outline	9/12/02	9/12/02	9/12/02
.5 Initial Full Draft	10/22	10/22	10/22
.7 1st F2F input inc. Format Complete	12/13	12/13	12/13
.8 Functionally Done Few open Issues	1/17/03	2/12/03	2/19/03
.85 Feedback Included Consistency Checks	-	-	-
.9 Essentially Complete Ready for Internal Reviews	2/14	3/7	3/7
.95 30 day Copyright Review	3/7	4/11	4/11
1.0 Release to Web/IETF	4/14	5/16	5/16

RSTG-mtg-030317-pat.doc

HP moves that –

A. The Send_Control invocation for sending a SCSI command includes the additional qualifier "UnsolicitedDataSize", that indicates the size of immediate and non-immediate unsolicited data for the command.

B. iSER spec should require an initiator to allocate an STag for a write I/O buffer (in SCSI Writes or bidirectional commands) and Advertise it to the target only if the Expected Data Transfer Length of the SCSI command > UnsolicitedDataSize.

C. iSCSI layer on the target MUST NOT issue recovery R2Ts on an iSCSI/iSER connection for a task whose connection allegiance was never reassigned (the only plausible reason being a sequence reception timeout in such a case). The target iSER layer may reject such a recovery R2T received via the Get_Data primitive, from the target iSCSI layer, with an appropriate error code. iSCSI layer on the target MAY however issue recovery R2Ts on an iSCSI/iSER connection, if the related task was earlier reassigned ([iSCSI]) to that connection.

Intel seconds

Adaptec	Abs
Agilent	N
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Network Appliance	--

Motion passes.

RSTG-mtg-030331.doc

IBM moves that we add a qualifier on the Put_Data and Get_Data operational primitives which, if set, signifies that the Data_Completion_Notify operational primitive MUST be driven upon completion. If not set, signifies that the Data_Completion_Notify operational primitive MUST NOT be driven upon completion.

Intel seconds.

Adaptec	Y
Agilent	Y
Broadcom:	Abstain
EMC	
HP	N
IBM	Y
Intel	Y
Network Appliance	

Motion passes.

IBM moves that we change the schedule of version 0.9 of iSER and DI specs to 4/4/2003 and change the 0.95 version to 5/16/2003 and the version 1.0 to 6/16/2003

HP seconds
Objections - None
Abstentions - None
Motion passes

RSTG-mtg-030402.doc

IBM motions "The feature set for the iSER/DI spec is frozen as of 4/2/03. Any new feature additions will require a two thirds majority vote to be adopted."

Broadcom Seconds

Adaptec	Y
Agilent	Y
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Network Appliance	absent

Motion passed.

RSTG-mtg-030424-mk.doc

HP motions that the DI 0.9 document be enhanced to be an architecture document and be called Datamover Architecture document (DA). iSER will be an instantiation of this architecture.

Adaptec seconds
Objections - None
Abstentions - None
Motion passes

RSTG-mtg-030429-tt.doc

IBM moves that we add the change to require that the target's iSER-ORD be reduced to be less than or equal to the initiator's iSER-IRD as declared in the iSER Hello message, and alter the appropriate text to use iSER-ORD instead of minimum of IRD and ORD. Include informational text on how to derive iSER-IRD and iSER-ORD from the RI.

EMC seconds
Objections - None
Abstentions - None
Motion passes

RSTG-mtg-030514.doc

IBM moves to forward the DA 0.95 and iSER 0.95 specifications to the Contributors's working group.

HP Seconds

Adaptec	Y
Agilent	Y
Broadcom:	Absent
EMC	Y
HP	Y
IBM	Y
Intel	Y
Network Appliance	Y

Motion passed.

RSTG-mtg-030616.doc

IBM moves that we approve the changes made to address the errata for iSER/DA version 0.95 release.

HP seconds

Adaptec	Y
Agilent	Y
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Network Appliance	Y

Motion passed unanimously.

IBM moves that we remove the change bar from DA version 0.95 and iSER version 0.97 and create version 0.98 to be forwarded to the Contributor's working group as a candidate for version 1.0 of DA and iSER specifications.

HP seconds

Adaptec	Y
Agilent	Y
Broadcom:	Y
EMC	Y
HP	Y
IBM	Y
Intel	Y
Network Appliance	Y

Motion passed unanimously.

1. The iSCSI data acknowledgment model

The iSCSI protocol specification defines a data acknowledgment model that is primarily intended for iSCSI targets to efficiently manage their buffers in responding to SCSI Read commands. The idea is that by proactively seeking a data acknowledgment from the iSCSI layer on the initiator and receiving it, the iSCSI layer on the target can be certain that the just acknowledged data will not be requested for a retransmission down the road by the iSCSI layer on the initiator. This assurance could greatly aid the iSCSI and SCSI layers on the target because the data buffers don't have to be kept around until the end of the task, but can be immediately reused for other I/O demands.

The iSCSI layer on the target conveys the data acknowledgment request to the initiator iSCSI layer via the setting of the A-bit in the iSCSI Data-in PDU from the target. Upon receiving a valid data acknowledgment request via the A-bit, the initiator iSCSI layer in turn is required to respond back with a data acknowledgment response in the form of an iSCSI SNACK PDU. This request-response interaction thus will acknowledge all the data up to and including the Data-in PDU that had the A-bit set.

Note that the whole iSCSI data acknowledgment model is usable only if the operational ErrorRecoveryLevel of the iSCSI session is greater than 0. An ErrorRecoveryLevel of 0 means that there's no data recovery, so a data acknowledgment is not useful by definition. An ErrorRecoveryLevel of 1 means that data recovery is supported on that iSCSI session, so the iSCSI layer on the target could deploy data acknowledgments to realize some efficiency in local buffer management. An ErrorRecoveryLevel of 2 implies task failover is also supported in addition to all that of ErrorRecoveryLevel=1, so the iSCSI layer on the target may still use data acknowledgments for the same reason.

2. The RDMA Datamover model

In the RDMA Datamover model (such as the one employed by the iSER protocol running on the iWARP protocol suite), the Datamover layers on the initiator and the target are responsible for moving all the data unbeknownst to the iSCSI layer on the initiator. In the case of a SCSI Read command, the iSER layer on the target simply moves all the Read data using one or more RDMA Writes back to the initiator RNIC. The key benefit of this data movement model is that the iSCSI layer on the initiator receives only a single interrupt at the conclusion of any SCSI Command (a Read command in this scenario), because it is not involved in the data movement.

3. Data acknowledgment problems with the RDMA model

As much as the RDMA Datamover model is compelling, it poses certain problems for the data acknowledgment expectations of the iSCSI specification. The biggest problem is that the iSCSI layer on the initiator cannot acknowledge any data back to the target iSCSI layer since the initiator iSCSI layer was not involved in the data movement to begin with! The second biggest problem is that the iSCSI layer on the initiator needs to be interrupted in order to respond to the acknowledgment request, and this violates the single interrupt per I/O model. This multiple interrupt problem is further compounded for the initiators because there is no limit on the number of acknowledgments that may be sought during the course of a SCSI Read command. Thus the number of interrupts an initiator may have to field for a long-running I/O is unbounded.

4. Proposed solution

If the data acknowledgment request originated by the iSCSI target layer can be couched in a generic form (i.e. without iSCSI or iSER specifics) so that it may be automatically responded to by the initiator's RDMA-capable Network Interface Controller (RNIC), it solves the problems mentioned in 3.

An RDMA Read Request is a good "generic form" of an acknowledgement request. An RDMA Read Request-Read Response pair is analogous in an abstract sense to a data acknowledgment request-response pair, and the RNIC receiving the RDMA Read Request on the *initiator* can automatically respond (without generating a local interrupt to the iSER layer) with an RDMA Read Response. Hence the proposal is to use an RDMA Read Request to "read" zero bytes out of the initiator's memory, whenever the A-bit is set on a Data-in PDU coming down from the iSCSI layer on the target. The RDMA Write/Read ordering rules of the RDMA Protocol ensure that the RDMA Read Request will not pass the RDMA Write Request and so the RDMA Read Request essentially acts to "flush" the connection of all the preceding RDMA Writes carrying the SCSI Data-in. The iSER layer on the target, when it receives the RDMA Read Response, can generate a notification back to the local iSCSI layer notifying it of the arrival of the data acknowledgment, essentially mimicking the SNACK-based data acknowledgment response.

It turns out that there is a further optimization possible here based on the operational ErrorRecoveryLevel of the iSCSI session. This optimization will simplify the wire protocol and iSER layer-to-RNIC interactions on the *target* in realizing the iSCSI data acknowledgment. If the operational ErrorRecoveryLevel is 1, as described in section 1, the connection recovery feature of the iSCSI protocol cannot be used. This further implies that if the connection fails for any reason, the data associated with the tasks on the failed connection will not be requested on a new connection since the tasks cannot be failed over. The RNIC interface guarantees that once the completion message for an RDMA operation is delivered to the iSER layer, the local data buffers associated with that RDMA operation will not be accessed by the RNIC and the associated data will be transferred if the connection stays up,. The combined implication is that when the operational ErrorRecoveryLevel=1, the iSER layer on the target can simply mimic a data acknowledgment response (as if received from the initiator) based on the RNIC-local completion message of the RDMA Write operation associated with the SCSI Data-in.

The pseudo-code of the proposed algorithm thus would look as describe din section 5.

5. Algorithm for the target iSER layer

If (the A-bit is set on the SCSI Data-in PDU) then
    If (the operational ErrorRecoveryLevel=2 ***or*** *if ErrorRecoveryLevel is unknown) then*
        Generate the standard RDMA Write for the SCSI Data-in PDU.
        Generate a zero-length RDMA Read Request after the RDMA Write.
        Wait for the RDMA Read Response arrival
    else if (the operational ErrorRecoveryLevel=1) then
        Generate the standard RDMA Write for the SCSI Data-in PDU.
        Wait for the local RDMA Write Completion
    endif
endif

Once the event being waited for – RDMA Read Response arrival or the local RDMA Write completion – occurs, the iSER layer on the target must generate a data acknowledgment notification to the iSCSI layer. This completes the iSCSI data acknowledgment expectations as far as the target iSCSI layer is concerned. Note that the initiator iSER or iSCSI layers need no special handling or logic in this proposed model.

6. Conclusion

This paper discusses the problems faced in meeting the iSCSI data acknowledgment expectations in the context of an RDMA Datamover and proposes a solution to the problem. The proposed solution preserves the "single interrupt for SCSI command" model on the initiator, even while meeting the data acknowledgment needs of the iSCSI layer on a target. The proposed solution in addition also includes a performance optimization on the target side that will cut down the wire protocol exchanges and speeds up the data acknowledgment response back to the iSCSI layer.